Journal of Information and Data Management, 2024, 15:1, doi: 10.5753/jidm.2024.3463

Instance hardness measures for classification and regression problems

Gustavo P. Torquette [**Universidade Federal de São Paulo** | ***gustavo.torquette@unifesp.br***]
Victor S. Nunes [**UNIFESP and Instituto Tecnológico de Aeronáutica** | ***victor.nunes@ga.ita.br***]
Pedro Y. A. Paiva [**Instituto Tecnológico de Aeronáutica** | ***paiva@ita.br***]
Ana C. Lorena [**Instituto Tecnológico de Aeronáutica** | ***aclorena@ita.br***]

Instituto Tecnológico de Aeronáutica, Praça Marechal Eduardo Gomes, 50, São José dos Campos, SP, 12228-900, Brazil.

Received: 15 June 2023 • **Published:** 27 February 2024

Abstract While the most common approach in Machine Learning (ML) studies is to analyze the performance achieved on a dataset through summary statistics, a fine-grained analysis at the level of its individual instances can provide valuable information for the ML practitioner. For instance, one can inspect whether the instances which are hardest to have their labels predicted might have any quality issues that should be addressed beforehand; or one may identify the need for more powerful learning methods for addressing the challenge imposed by one or a set of instances. This paper formalizes and presents a set of meta-features for characterizing which instances of a dataset are the hardest to have their label predicted accurately and why they are so, aka instance hardness measures. While there are already measures able to characterize instance hardness in classification problems, there is a lack of work devoted to regression problems. Here we present and analyze instance hardness measures for both classification and regression problems according to different perspectives, taking into account the particularities of each of these problems. For validating our results, synthetic datasets with different sources and levels of complexity are built and analyzed, indicating what kind of difficulty each measure is able to better quantify. A Python package containing all implementations is also provided.

Keywords: Data complexity, Instance Hardness, Hardness Measures, Machine Learning

1 Introduction

Data is a crucial component for the development of Machine Learning (ML) models, making endeavours towards better understanding data properties important [Schweighofer, 2021]. By properly characterizing data one may previously identify quality issues that need to be addressed [Paiva *et al.*, 2022] or the need for more powerful solutions for solving a new problem [Cruz *et al.*, 2015, 2018; Sowkarthika *et al.*, 2023].

Some recent piece of work has been focusing on characterizing a dataset based on the difficulty level encountered in predicting the labels of its individual instances, a concept named *instance hardness* analysis [Smith *et al.*, 2014; Arruda *et al.*, 2020; Torquette *et al.*, 2022]. Objectively, Smith *et al.* [2014] regard an instance as hard to classify if a set of predictors of different biases are unable to predict its label correctly. This concept is originally defined for classification problems, by taking the average probability of misclassification of an instance as recorded by a pool of diverse classification algorithms. For regression problems, where the data labels are quantitative, a first strategy for instance hardness quantification was proposed in [Torquette *et al.*, 2022]. Similarly to what is performed in classification problems, the predictions of a pool of diverse regressors for the instance are taken into account. Nonetheless, in this case, the distance between the predictions and the actual label values is taken.

In addition to the identification of instances which are the hardest to have their labels predicted, the literature also contains measures which give estimates of instance hardness quantification according to different perspectives [Smith *et al.*, 2014; Arruda *et al.*, 2020; Torquette *et al.*, 2022]. They are named *instance hardness measures* (IHM) and are aimed to explain possible reasons why an instance is hard in a dataset. In classification problems, instances near the decision boundary needed to separate the classes or which are outliers tend to have higher hardness levels than instances well situated inside their classes [Lorena *et al.*, 2019]. In regression problems, outliers also pose a higher complexity and the smoothness of the data distribution can influence the predictive results attained too [Lorena *et al.*, 2018]. There are several IHM defined for classification problems in the literature, but the analysis of regression problems is still incipient.

Other attempts to measure the hardness level of individual instances in a dataset are given by the use of concepts from Item Response Theory (IRT) [Martínez-Plumed *et al.*, 2019; Moraes *et al.*, 2022]. In this framework, each instance of the dataset is regarded as an item of an exam or test, while the respondents of the test are different ML models. Based on the pool of responses, it is possible to identify latent traits such as the difficulty level of the items (instances) and the ability level of the classifiers.

This work focuses on estimating instance hardness with the support of the IHM. They are able to give estimates of the instance hardness level without the need to run multiple ML models, which can be costly. They also present different perspectives on why an instance is hard to predict in comparison to others in a dataset. IHM for classification and regression problems are presented, formalized and experimentally analysed. Finally, we provide a Python package containing implementations of all IHM presented.

The experiments comprise a set of synthetic datasets with increasing sources and levels of difficulty. The choice for synthetic datasets is motivated by the need to control the data behavior, which would be more difficult to achieve using real datasets. Our results evidence how each IHM contributes in identifying different sources of classification/regression difficulty. This paper extends our previous contribution on the topic [Torquette *et al.*, 2022], which was limited in the formalization of the measures and their experimentation. In this extended paper, we present the mathematical formalization of the measures absent in the previous contribution. The experiments here also include more datasets, which now consider a wider range of possible sources of difficulty of classification and regression problems to be captured, making the contribution more complete.

This paper is structured as follows: Section 2 presents the background on measuring instance hardness. Section 3 presents the IHM of classification and regression problems. Section 4 presents an experimental evaluation of the IHM. Section 5 concludes this paper.

2 Instance Hardness Analysis

The most common approach when analyzing a new dataset is giving aggregate estimates on its main properties [Vanschoren, 2019; Rivolli *et al.*, 2022; Lorena *et al.*, 2019]. Smith *et al.* [2014] was seminal to stress the need to perform a more fine-grained analysis of a dataset, by investigating the difficulty level in predicting the label of each individual instance of a dataset.

Formally, given a dataset $\mathcal{D}$ containing n instances $\mathbf{x}_i \in X$ with their corresponding labels $y_i \in Y$, *instance hardness* (IH) is defined as the likelihood of an instance being misclassified by a pool of classifiers $\mathcal{L}$ with different biases when trained on $\mathcal{D}$:

$$\mathrm{IH}_{\mathcal{L}}(\mathbf{x}_i, y_i) = 1 - \frac{1}{|\mathcal{L}|}\sum_{j=1}^{|\mathcal{L}|} p\big(y_i | h_j(\mathbf{x}_i)\big), \quad (1)$$

where $p\big(y_i|h_j(\mathbf{x}_i)\big)$ is the probability $\mathbf{x}_i$ is assigned to its original label y_i by a learning model h_j from the pool. The concept is originally defined for classification problems, where the labels y_i are qualitative (Y is a discrete set). The idea is that instances that are frequently misclassified by a pool of diverse learning algorithms can be considered hard to classify. In contrast, easy instances are likely to be correctly classified by any of the algorithms.

In regression problems, the quantitative nature of the output labels (set Y) prevents a direct quantification of the probability in Equation 1. Nonetheless, it is a fact that the closer the predicted value is to the actual label of an instance, the more accurate the regressor response is. Therefore, it is more intuitive to define our probability space for regression problems over distances $z = d\,(y_i, h(\mathbf{x}_i))$, where y_i is the label of an instance $\mathbf{x}_i$ and $h(\mathbf{x}_i)$ is the prediction obtained by a model h. Taking an exponential kernel on such distances and plugging it into the cumulative distribution function, we arrive in the form of the exponential distribution. Now, we can define the instance hardness for regression as:

$$IH_{\mathcal{L}}(\mathbf{x}_i, y_i) = 1 - \frac{1}{|\mathcal{L}|}\sum_{j=1}^{|\mathcal{L}|} \exp\left(-\frac{d\,(y_i, h_j(\mathbf{x}_i))}{\gamma}\right), \quad (2)$$

where $h_j(\mathbf{x}_i)$ represents the output of a regressor j in the pool $\mathcal{L}$ for instance $\mathbf{x}_i$. A natural issue that arises is how to choose a proper value for γ, which can be seen as a normalizing constant. For instance, γ can be set as the power of the signal $\mathbf{y}$, that is, $\gamma = \frac{1}{n}\sum_i y_i^2$. In that case, depending on the chosen distance metric $d(\cdot,\cdot)$, $\frac{d(\cdot,\cdot)}{\gamma}$ is equivalent to some normalized error metric, such as normalized squared error when the the Euclidean distance is used.

For computing the IH values in Equations 1 and 2, a cross-validation procedure is undertaken. $\mathcal{D}$ is divided into r folds of approximately the same size and while $r-1$ folds are used for training, one is left out for testing. Each training instance is part of one of the test folds, making it possible to compare the instance actual label to the predictions obtained by a pool of models and to compute the IH values.

3 Instance Hardness Measures (IHM)

The *instance hardness measures* (IHM) allow to obtain estimates of IH without the need to rely on the results of multiple ML models, while also indicating possible reasons why an instance is hard to be predicted. Next we present the IHM surveyed in this work, along with their formal definition and interpretation.

All of them are computed for the instances of a dataset $\mathcal{D}$ with n data instances $\mathbf{x}_i$ assuming labels y_i in a set Y, which is qualitative for classification problems and quantitative for regression problems. In addition, let m denote the number of input features the dataset has. The definitions of all measures are standardized so that larger values are observed for more difficult instances.

3.1 IHM for Classification

The IHM for classification described here were originally proposed in [Smith *et al.*, 2014; Arruda *et al.*, 2020; Torquette *et al.*, 2022] and are formalized next. Other measures with similar concepts are omitted. Some measures have an IHM subscript in order to differentiate them from their counterparts defined at an aggregated dataset level in [Lorena *et al.*, 2019].

k-Disagreeing Neighbors $\mathrm{kDN}(\mathbf{x}_i)$: outputs the percentage of the k nearest neighbors of $\mathbf{x}_i$ in $\mathcal{D}$ which do not share its label:

$$\mathrm{kDN}(\mathbf{x}_i) = \frac{\sharp\{\mathbf{x}_j | \mathbf{x}_j \in \mathrm{kNN}(\mathbf{x}_i) \wedge y_j \neq y_i\}}{k}, \quad (3)$$

where $\mathrm{kNN}(\mathbf{x}_i)$ represents the set of k-nearest neighbors of the instance $\mathbf{x}_i$ in the dataset $\mathcal{D}$, being k frequently set to 5 [Smith *et al.*, 2014]. The higher the value of kDN($\mathbf{x}_i$), the harder $\mathbf{x}_i$'s classification tends to be, since it is surrounded by examples from a different class.

Disjunct Class Percentage DCP($\mathbf{x}_i$): this IHM builds a pruned decision tree (DT) using $\mathcal{D}$ and considers the percentage of instances in the disjunct of $\mathbf{x}_i$ which share the same label as $\mathbf{x}_i$, where the disjunct of an instance corresponds to the leaf node where it is classified by the DT:

$$\text{DCP}(\mathbf{x}_i) = 1 - \frac{\sharp\{\mathbf{x}_j | \mathbf{x}_j \in \text{Disjunct}(\mathbf{x}_i) \wedge y_j = y_i\}}{\sharp\{\mathbf{x}_j | \mathbf{x}_j \in \text{Disjunct}(\mathbf{x}_i)\}}, \tag{4}$$

where Disjunct($\mathbf{x}_i$) represents the set of instances contained in the disjunct (leaf node) where $\mathbf{x}_i$ is placed. Easier instances will have a larger percentage of examples sharing the same label as them in their disjunct, so we output the complement of this percentage.

Tree Depth TD($\mathbf{x}_i$): gives the depth of the leaf node that classifies $\mathbf{x}_i$ in an unpruned decision tree, normalized by the maximum depth of the tree built from $\mathcal{D}$:

$$\text{TD}(\mathbf{x}_i) = \frac{\text{depth}_{\text{DT}}(\mathbf{x}_i)}{\max(\text{depth}_{\text{DT}}(\mathbf{x}_j \in \mathcal{D}))}, \tag{5}$$

where depth$_{\text{DT}}(\mathbf{x}_i)$ gives the depth where the instance $\mathbf{x}_i$ is placed in the DT. There is also a version of this measure in which the DT is pruned, but we did not include here because it is very correlated to the unpruned version. Harder to classify instances tend to be placed at deeper levels of the tree and present higher TD values.

Class Likelihood Difference CLD($\mathbf{x}_i$): takes the difference between the likelihood that $\mathbf{x}_i$ belongs to its class y_i and the maximum likelihood it has to any other class. The difference in the class likelihood is larger for easier instances, because the confidence it belongs to its class is larger than that of any other class. We take the complement of the difference for standardizing the interpretations of the hardness directions:

$$\text{CLD}(\mathbf{x}_i) = \frac{1 - \left(p(\mathbf{x}_i|y_i)p(y_i) - \max_{y_j \neq y_i}[p(\mathbf{x}_i|y_j)p(y_j)]\right)}{2}, \tag{6}$$

where $p(\mathbf{x}_i|y_i)$ represents the likelihood $\mathbf{x}_i$ belongs to class y_i and $p(y_i)$ is the prior of class y_i, which we set as $\frac{1}{C}$ for all data instances, where C is the number of classes. The conditional probability $p(\mathbf{x}_i|y_i)$ can be estimated considering each of the input features independent from each other, as in Naïve Bayes classification.

Fraction of features in overlapping areas F1$_{\text{IHM}}(\mathbf{x}_i)$: this measure takes the percentage of features of the instance $\mathbf{x}_i$ whose values lie in an overlapping region of the classes using:

$$\text{F1}_{\text{IHM}}(\mathbf{x}_i) = \frac{\sum_{j=1}^{m} I(x_{ij} \geq \text{max\_min}(\mathbf{f}_j) \wedge x_{ij} \leq \text{min\_max}(\mathbf{f}_j))}{m}, \tag{7}$$

where I is the indicator function, which returns 1 if its argument is true and 0 otherwise, $\mathbf{f}_j$ is the j-th feature vector in $\mathcal{D}$ and:

$$\text{min\_max}(\mathbf{f}_j) = \min(\max(\mathbf{f}_j^{c_1}), \max(\mathbf{f}_j^{c_2})), \tag{8}$$

$$\text{max\_min}(\mathbf{f}_j) = \max(\min(\mathbf{f}_j^{c_1}), \min(\mathbf{f}_j^{c_2})). \tag{9}$$

The values $\max(\mathbf{f}_j^{y_i})$ and $\min(\mathbf{f}_j^{y_i})$ are the maximum and minimum values of $\mathbf{f}_j$ in a class $y_i \in \{c_1, c_2\}$. Therefore, the overlap for a feature $\mathbf{f}_j$ is measured according to the maximum and minimum values it assumes in two different classes. One may regard a feature as having overlap if it is not possible to separate the classes using a threshold on that feature's values. F1$_{\text{IHM}}$ gives the percentage of features for which an example is in an overlapping region according to this definition. Larger values of F1$_{\text{IHM}}$ are obtained for data instances which lie in overlapping regions for most of the features, implying they are harder to classify according to the F1$_{\text{IHM}}$ interpretation. Multiclass classification problems must be first decomposed into multiple binary classification problems. Different strategies can be used in this decomposition, such as one-vs-all (OVA - one class against the others) or one-vs-one (OVO - each pairwise combination of the classes) [Lorena *et al.*, 2008]. In our implementation, we have opted for OVO, because the generated subproblems tend to be smaller and with a more even distribution of the classes.

Fraction of nearby instances of different classes N1$_{\text{IHM}}(\mathbf{x}_i)$: in this measure, first a minimum spanning tree MST is built, where each instance of the dataset $\mathcal{D}$ corresponds to one vertex and nearby instances are connected according to their distances in order to obtain a tree of minimal cost concerning the sum of the edges' weights. N1$_{\text{IHM}}$ gives the percentage of instances of different classes $\mathbf{x}_i$ is connected to in the MST:

$$\text{N1}_{\text{IHM}}(\mathbf{x}_i) = \frac{\sharp\{\mathbf{x}_j | (\mathbf{x}_i, \mathbf{x}_j) \in \text{MST}(\mathcal{D}) \wedge y_i \neq y_j\}}{\sharp\{\mathbf{x}_j | (\mathbf{x}_i, \mathbf{x}_j) \in \text{MST}(\mathcal{D})\}}. \tag{10}$$

Larger values indicate that $\mathbf{x}_i$ is close to examples of different classes, either because it is borderline or noisy, making it hard to classify.

Ratio of the intra-class and extra-class distances N2$_{\text{IHM}}(\mathbf{x}_i)$: takes the complement of the ratio of the distance of $\mathbf{x}_i$ to the nearest example from its class to the distance it has to the nearest instance from a different class (nearest enemy - NE):

$$\text{N2}_{\text{IHM}}(\mathbf{x}_i) = 1 - \frac{1}{\text{IntraInter}(\mathbf{x}_i) + 1}, \tag{11}$$

where:

$$\text{IntraInter}(\mathbf{x}_i) = \frac{d(\mathbf{x}_i, \text{NN}(\mathbf{x}_i) \in y_i)}{d(\mathbf{x}_i, \text{NE}(\mathbf{x}_i))}, \tag{12}$$

where NN($\mathbf{x}_i$) represents a nearest neighbor of $\mathbf{x}_i$ and NE($\mathbf{x}_i$) is the nearest enemy of $\mathbf{x}_i$ (NE($\mathbf{x}_i$) $=$ NN($\mathbf{x}_i$) $\in y_j \neq y_i$). Larger values of N2$_{\text{IHM}}$ indicate that $\mathbf{x}_i$ is closer to an example from another class than to an example from its own class, making it harder to classify.

Local Set Cardinality LSC$_{\text{IHM}}(\mathbf{x}_i)$: the Local-Set (LS) of an instance $\mathbf{x}_i$ is the set of points from class y_i in $\mathcal{D}$ whose distances to $\mathbf{x}_i$ are smaller than the distance between $\mathbf{x}_i$ and $\mathbf{x}_i$'s nearest enemy [Leyva *et al.*, 2014]:

$$LS(\mathbf{x}_i) = \sharp\{\mathbf{x}_j | d(\mathbf{x}_i, \mathbf{x}_j) < d(\mathbf{x}_i, \text{NE}(\mathbf{x}_i))\}, \tag{13}$$

LSC$_{\text{IHM}}$ outputs the complement of the relative cardinality of such set:

$$\text{LSC}_{\text{IHM}}(\mathbf{x}_i) = 1 - \frac{|LS(\mathbf{x}_i)|}{\sharp\{\mathbf{x}_j|y_i = y_j\}}. \quad (14)$$

Larger local sets are obtained for easier examples, which are in dense regions surrounded by instances sharing their class labels. For standardization, we output a complement of the relative local set cardinality.

Local Set Radius LSR$(\mathbf{x}_i)$: takes the normalized radius of $\mathbf{x}_i$'s local set:

$$\text{LSR}(\mathbf{x}_i) = 1 - \min\left\{1, \frac{d(\mathbf{x}_i, \text{NE}(\mathbf{x}_i))}{\max(d(\mathbf{x}_i, \mathbf{x}_j)|y_i = y_j)}\right\} \quad (15)$$

Larger radiuses are expected for easier instances, which are surrounded by many instances from their class, so we take the complement of such measure.

Usefulness U$(\mathbf{x}_i)$: corresponds to the fraction of instances having $\mathbf{x}_i$ in their local sets [Leyva *et al.*, 2015]:

$$\text{U}(\mathbf{x}_i) = 1 - \frac{\sharp\{\mathbf{x}_j|d(\mathbf{x}_i, \mathbf{x}_j) < d(\mathbf{x}_j, \text{NE}(\mathbf{x}_j))\}}{\sharp\{\mathbf{x}_j|y_j = y_i\}} \quad (16)$$

If $\mathbf{x}_i$ is easy to classify, it will be close to many examples from its class and therefore will be more useful. We take the complement of this measure as output.

Harmfulness H$(\mathbf{x}_i)$: is the number of instances having $\mathbf{x}_i$ as their nearest enemy [Leyva *et al.*, 2015]:

$$\text{H}(\mathbf{x}_i) = \frac{\sharp\{\mathbf{x}_j|\text{NE}(\mathbf{x}_j) = \mathbf{x}_i\}}{\sharp\{\mathbf{x}_j|y_j \neq y_i\}} \quad (17)$$

If $\mathbf{x}_i$ is the nearest enemy of many instances, this indicates it is harder to classify and its harmfulness will be higher.

Degree centrality Degree$_{\text{IHM}}(\mathbf{x}_i)$: this measure is based on a complexity measure originally taken at a dataset-level [Lorena *et al.*, 2019], which models the dataset as a proximity graph. First a graph $G = (V, E)$ is built from $\mathcal{D}$, connecting pairs of instances from the same class for which the distance is inferior to a threshold ϵ, set as 15% of the smallest distances, as in [Morais and Prati, 2013; Garcia *et al.*, 2015]. This graph presents at least C connected components, one for each class, connecting elements from a same class which are similar to each other. The complement of the density of the connections a vertex v_i has in the graph gives its hardness level:

$$\text{Degree}_{\text{IHM}}(\mathbf{x}_i) = 1 - \frac{E(v_i)}{\sharp\{\mathbf{x}_j|y_j = y_i\} - 1}, \quad (18)$$

where $E(v_i)$ the number of edges of the vertex corresponding to $\mathbf{x}_i$.

If $\mathbf{x}_i$ is easy to classify, it will be surrounded by close elements from its class and will have a lower Degree value as measured by Equation 18.

Closeness centrality Closeness$_{\text{IHM}}(\mathbf{x}_i)$: this measure is based on the same graph G built previously. The closeness centrality measure of a vertex $v_i \in G$ is the reciprocal of the sum of the length of the shortest paths between the vertex and all other vertices of the graph. The more central a vertex (instance) is in the graph, the closer it is to all other nodes.

$$\text{Closeness}_{\text{IHM}}(\mathbf{x}_i) = 1 - \frac{\sharp\{\mathbf{x}_j|y_j = y_i\} - 1}{\sum_{v_j \in y_i} d(v_i, v_j)}, \quad (19)$$

where v_i and v_j are vertices of the network and $d(v_i, v_j)$ is the distance between these two vertices. The closeness measure as previously defined returns lower values for instances in regions containing a high density of points of their class.

3.2 IHM for Regression

The IHM for regression are based on complexity measures and meta-features for regression problems taken at the dataset-level [Lorena *et al.*, 2018], which are decomposed here at the instance-level.

Collective Feature Efficiency CFE$(\mathbf{x}_i)$: this measure starts by identifying the feature with highest correlation to the output vector in $\mathcal{D}$. All examples with a small residual value ($|\varepsilon_i| \leq 0.1$) after a linear fit between this feature and the target attribute are removed. Then, the most correlated feature to the remaining data points is found and the previous process is repeated until all features have been analyzed or no example remains. For an instance $\mathbf{x}_i$, we take the round l_i where it is removed from the analysis, normalized by the maximum number of rounds:

$$\text{CFE}(\mathbf{x}_i) = \frac{l_i}{m}. \quad (20)$$

The l_i value can range between 1 and m, where m is the number of input features the dataset has. Higher CFE values are obtained for harder instances, which require more features to get a linear fit. The CFE values range in $\left[\frac{1}{m}, 1\right]$.

Absolute Error after Linear fit LE$(\mathbf{x}_i)$: first a statistical model of a Multiple Linear Regression is fit to $\mathcal{D}$. For each $\mathbf{x}_i$, a residual or error ε_i in relation to the actual output y_i can be measured and LE is given as:

$$\text{LE}(\mathbf{x}_i) = |\varepsilon_i|. \quad (21)$$

Larger values of this measure are attained for harder instances according to the interpretation of this measure, meaning they deviate much from a linear fit.

Output Distribution S1$_{\text{IHM}}(\mathbf{x}_i)$: As in N1$_{\text{IHM}}$ for classification, first a MST is generated from input data, where each instance corresponds to a vertex of the graph, while the edges are weighted according to the distance between the examples in the input space. The MST will greedily connect examples nearest to each other. Next S1$_{\text{IHM}}$ monitors whether the instances joined in the MST have similar output values. Lower values are obtained for simpler instances, who have similar outputs to their neighbors in the input space as represented in the MST.

As an instance $\mathbf{x}_i$ can have multiple neighbors in the MST, we take the average of the differences between their outputs:

$$\mathrm{S1}_{\mathrm{IHM}}(\mathbf{x}_i) = \frac{1}{\sharp\{\mathbf{x}_j|(\mathbf{x}_i,\mathbf{x}_j) \in \mathrm{MST}(\mathcal{D})\}} \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in \mathrm{MST}(\mathcal{D})} |y_i - y_j|, \tag{22}$$

where the denominator gives the number of neighbors of $\mathbf{x}_i$ in the MST. Higher values will be obtained for harder instances, which are connected to neighbors with more dissimilar outputs.

Input Distribution $\mathrm{S2}_{\mathrm{IHM}}(\mathbf{x}_i)$: $\mathrm{S2}_{\mathrm{IHM}}$ first orders the instances according to their output values y_i and then computes the Euclidean distance between pairs of examples that are neighbors. This measure complements $\mathrm{S1}_{\mathrm{IHM}}$ by measuring how similar in the input space are instances with similar outputs. Lower values $\mathrm{S2}_{\mathrm{IHM}}$ are obtained for simpler instances.

In the ordering, each element will either have one or two neighbor examples. For two neighbors, the average of the distances should be taken. Otherwise, the distance to the unique neighbor is output. Given that $y_1 \leq y_2 \leq \ldots \leq y_n$, that is, that the examples are already ordered according to their output values, we have:

$$\mathrm{S2}_{\mathrm{IHM}}(\mathbf{x}_i) = \begin{cases} d(\mathbf{x}_1, \mathbf{x}_2), & \text{for } i = 1 \\ d(\mathbf{x}_n, \mathbf{x}_{n-1}), & \text{for } i = n \\ \frac{d(\mathbf{x}_i,\mathbf{x}_{i-1})+d(\mathbf{x}_i,\mathbf{x}_{i+1})}{2}, & \text{otherwise.} \end{cases} \tag{23}$$

Squared Error of k-nearest neighbor $\mathrm{S3}_{\mathrm{IHM}}(\mathbf{x}_i)$: calculates the squared error (SE) of a *k-nearest neighbor regressor* (NN), using *leave-one-out*. As in kDN, the value of k is set to 5.

$$S3_{\mathrm{IHM}}(\mathbf{x}_i) = (k\mathrm{NN}(\mathbf{x}_i) - y_i)^2, \tag{24}$$

where $k\mathrm{NN}(\mathbf{x}_i)$ represents the k-nearest neighbor prediction for $\mathbf{x}_i$. Larger values are observed for harder instances.

The measures TD and Degree from the previous section can also be applied to regression problems. While TD will need a regression tree to be induced from $\mathcal{D}$ instead of the decision tree, Degree will take a proximity graph between the instances in $\mathcal{D}$, disregarding their outputs.

4 Experiments

In this section we perform experiments to show how the IHM behave for classification and regression datasets with different sources and levels of difficulty. All measures described previously are implemented in Python and distributed in the PyHard package library[1] [Paiva *et al.*, 2022].

4.1 Classification datasets

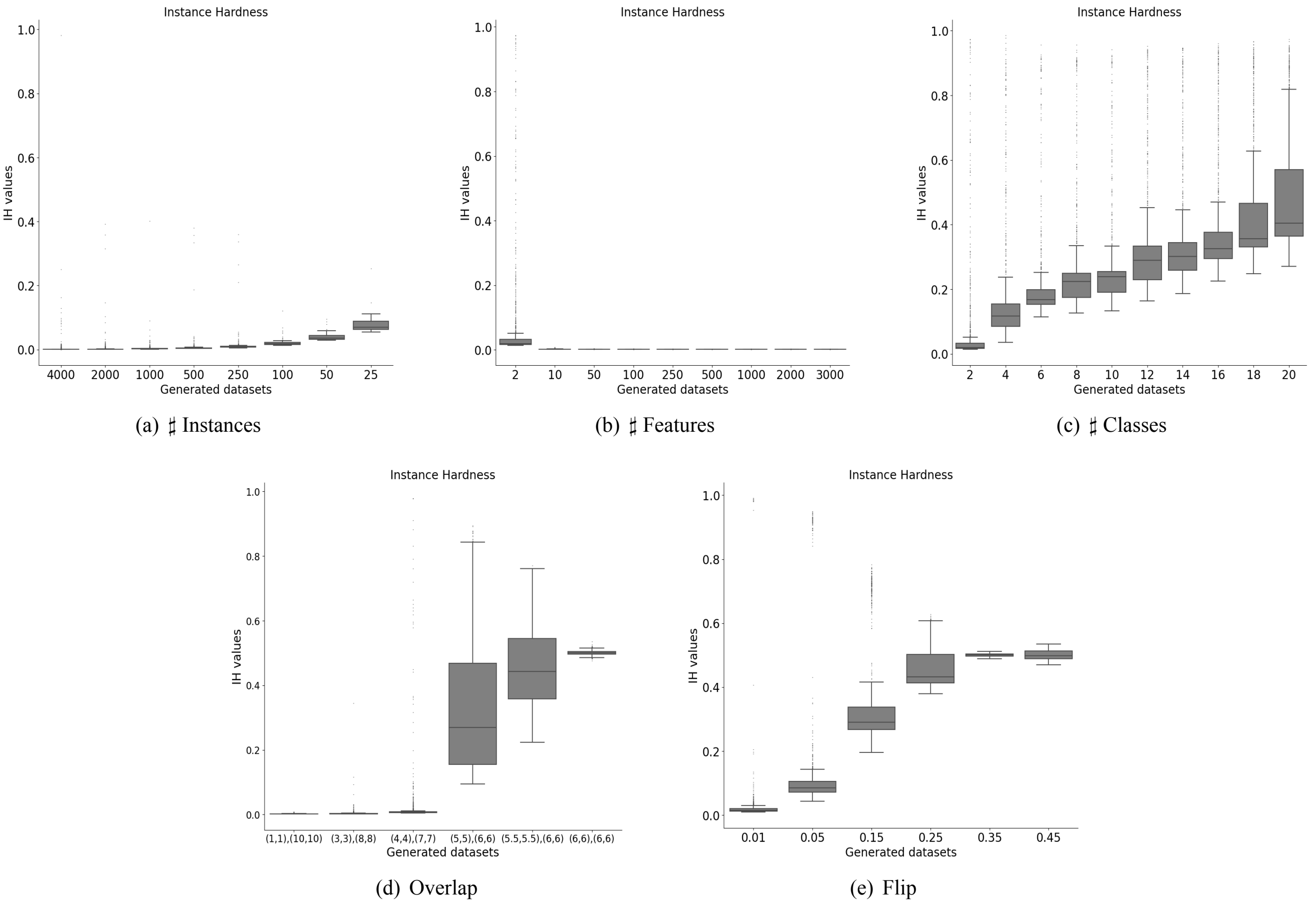
Using the `make_blobs` package from the scikit-learn library [Pedregosa *et al.*, 2011], we generated synthetic datasets where different characteristics which may affect the difficulty in solving a classification problem are varied. This package generates isotropic Gaussian blobs in the space, which can be regarded as classes. By adjusting one source of difficulty at a time, several datasets were generated, as follows:

1. **Number of instances**: datasets with two classes and two input features, varying the number of instances as: 25, 50, 100, 250, 500, 1000, 2000 and 4000. The greater the number of observations for a same number of input features, a lower level of difficulty in classifying the instances is expected.
2. **Number of features**: datasets with two classes and 1000 instances, varying the number of features as: 2, 10, 50, 100, 250, 500, 1000, 2000 and 3000. The greater the number of features for a same number of instances, the greater the expected level of difficulty, since the data tend to become sparse in the input space.
3. **Number of classes**: datasets with 1000 instances and two dimensions, with the “Centers” (number of groups or blobs, where each blob corresponds to one class) parameter varied as: 2, 4, 6, 8, 10, 12, 14, 16, 18 and 20. As the number of classes increases, a higher level of difficulty in classifying the instances correctly is expected.
4. **Overlap of the classes**: datasets with 1000 instances, two attributes and two classes, where the position of the centers of the classes is varied, making them to progressively overlap: [(1,1),(10,10)], [(3,3),(8,8)], [(4,4),(7 ,7)], [(5,5),(6,6)], [(5.5,5.5),(6,6)], [(6,6),(6,6)]. The more overlapped are the classes, a higher level of difficulty is expected.
5. **Label flip**: datasets with 1000 instances, two attributes and two classes and initially no overlap, where the labels of some instances are flipped at the following rates: 1%, 5%, 15%, 25%, 35% and 45%. The objective is to test the effects of a wrong data labeling in classification complexity, which is expected to increase.

We first verify how the instance hardness values behave for the different datasets' variants by using Equation 1 with the following pool $\mathcal{L}$ of classification techniques of distinct biases: Support Vector Machine (SVM) with linear Kernel, SVM with RBF Kernel, Random Forest (RF), Gradient Boosting (GB), Bagging, Logistic Regression and Multilayer Perceptron (MLP). All classification techniques were run in a 5-fold cross-validation procedure and had hyperparameters tuned by an inner 3-fold cross-validation on the training folds. Figure 1 presents the boxplots of the IH values for each set of datasets (instance hardness on the y axis and dataset in the x axis). For standardization purposes, the order of the datasets in the x axis is presented in expected increasing order of classification complexity. Therefore, in Figure 1a, the datasets are placed in decreasing order regarding the number of instances they contain (from the largest to the smallest).

Observing Figure 1, we may notice that some sources of classification difficulty affect more the performance of the ML techniques than others. For instance, as more overlapped the classes are, the higher tends to be the difficulty in classifying the instances. This also happens for higher flip rates, which ultimately also results in an overlap of instances of

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



(a) ♯ Instances

(b) ♯ Features

(c) ♯ Classes

(d) Overlap

(e) Flip

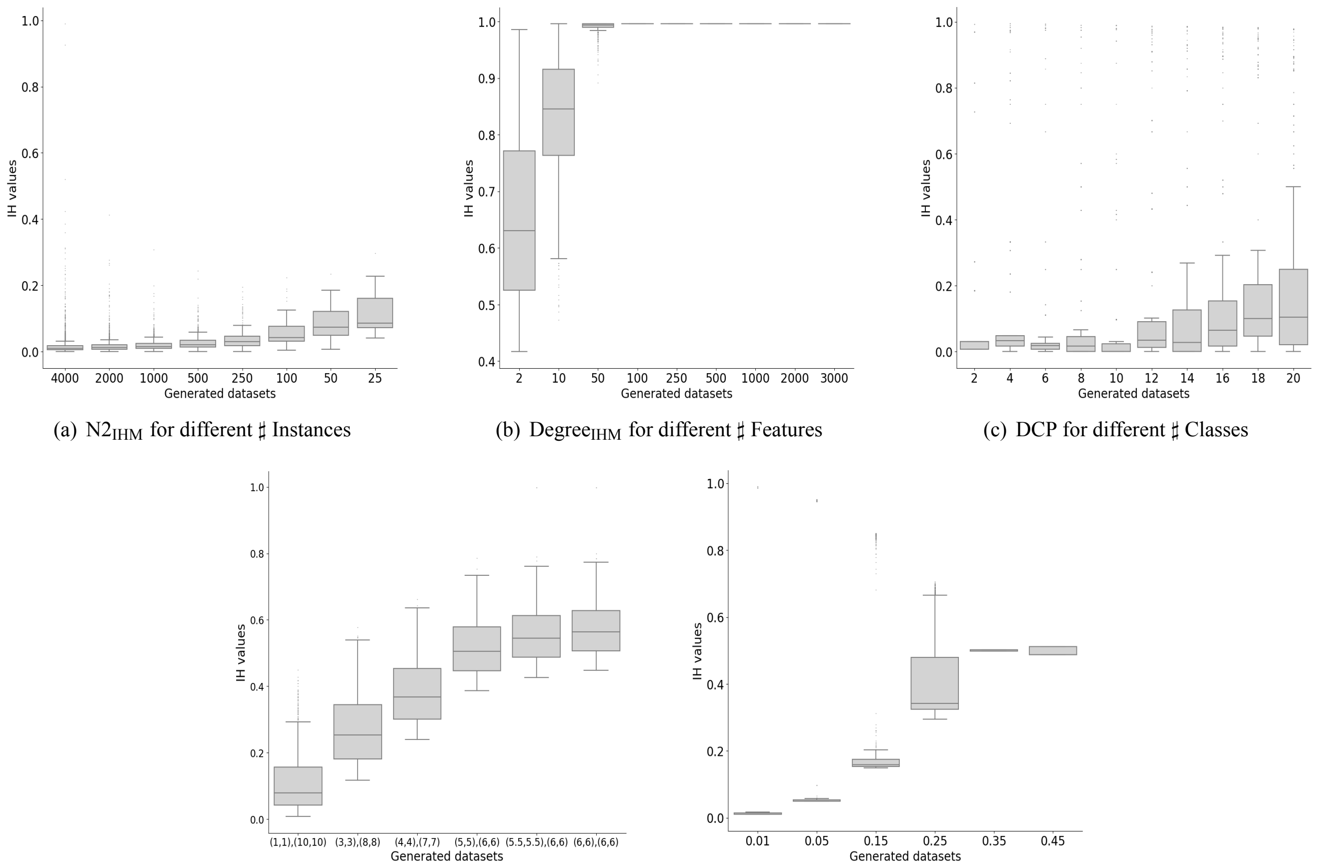
Figure 1. IH values measured at datasets of increasing complexities according to different difficulty sources.

different classes. Increasing the number of classes also leads to a steady increase of the instance hardness values. But decreasing the number of instances in our datasets did not influence much on the instance hardness observed, nor did increasing the number of input features, for which most of the instances remained easily classified.

Now we move our attention to how the IHM reflect the previous trends in instance hardness increase. For such, the previous datasets were input to the PyHard tool to obtain the IHM listed in Section 3.1. Figure 2 presents boxplots of some of the IHM values obtained for each set of datasets. Each plot presents the values of one IHM (y axis) along the different datasets (x axis) of increasing complexities. They all tend to increase their values for more complex datasets. For instance, datasets with less instances (Figure 2a) have lower N2_{IHM} values, meaning the intra-class distances become larger compared to the inter-class distances. When the number of input features increase while keeping the number of instances fixed, the dataset becomes more sparse, which reflects on an increase of the $\text{Degree}_{\text{IHM}}$ measure (Figure 2b). The larger the number of classes, the more the instances are mixed in disjuncts formed by DT on data, as reflected in Figure 2c for the DCP measure. The larger the overlap of the classes in Figure 2d, the $\text{Closeness}_{\text{IHM}}$ values become also higher, as there will be less connections between elements of the same class in the graph built from data. And the CLD values increase for higher flip rates (Figure 2e), as the likelihood an instance belongs to its class is low when its label is flipped into that of another class. But there are other combinations of IHM that are not so favorable towards representing an increased complexity level (the complete set of plots can be consulted at Appendix A).

For summarizing the relationship between the IHM and the instance hardness level of the instances in the datasets, Figure 3 presents a heatmap of the correlations (measured by Spearman correlation) between the IH as measured by a pool of classifiers (Equation 1) and the IHM values, for each of the types of datasets generated in this section. The higher the correlation, the redder the color, representing that as IH increases, the IHM values also increase. At a first glance, we can clearly see that the overlap is the type of difficulty captured more effectively by most of the IHM, with the exception of TD. Next comes the variation of the flip rate, which tends to create an overlap of the classes too. For the other sources of difficulty, the results are not so evident, except for CLD when the number of instances and the number of classes is varied. Indeed, CLD was one of the most effective measures in capturing the increase of difficulty of the instances in all of the scenarios tested, presenting high correlations to the IH values as measured by Equation 1. TD, in contrast, was not effective in capturing most of the sources of complexity tested in this paper and turned out to be more affected by the decrease in the number of instances and the increase in the number of classes of the datasets.

Although the previous analysis summarizes the overall behavior of the metrics, by examining the individual plots of



(a) $\mathrm{N2_{IHM}}$ for different ♯ Instances

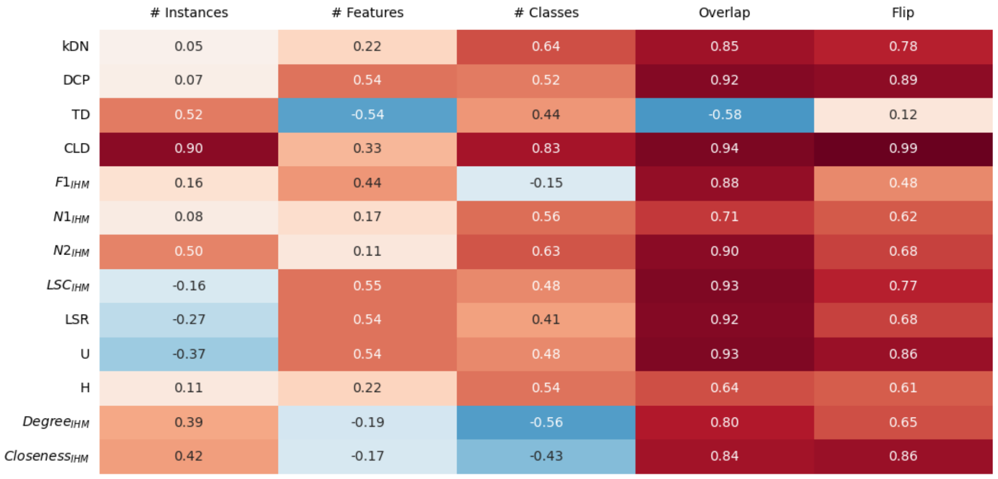
(b) $\mathrm{Degree_{IHM}}$ for different ♯ Features

(c) DCP for different ♯ Classes

(d) $\mathrm{Closeness_{IHM}}$ for different levels of overlap

(e) CLD for different flip rates

Figure 2. IHM values measured at datasets of increasing complexities according to different difficulty sources.



	# Instances	# Features	# Classes	Overlap	Flip
kDN	0.05	0.22	0.64	0.85	0.78
DCP	0.07	0.54	0.52	0.92	0.89
TD	0.52	-0.54	0.44	-0.58	0.12
CLD	0.90	0.33	0.83	0.94	0.99
$F1_{IHM}$	0.16	0.44	-0.15	0.88	0.48
$N1_{IHM}$	0.08	0.17	0.56	0.71	0.62
$N2_{IHM}$	0.50	0.11	0.63	0.90	0.68
LSC_{IHM}	-0.16	0.55	0.48	0.93	0.77
LSR	-0.27	0.54	0.41	0.92	0.68
U	-0.37	0.54	0.48	0.93	0.86
H	0.11	0.22	0.54	0.64	0.61
$Degree_{IHM}$	0.39	-0.19	-0.56	0.80	0.65
$Closeness_{IHM}$	0.42	-0.17	-0.43	0.84	0.86

Figure 3. Correlations between IHM and instance hardness measured by multiple classifiers for the different classification datasets generated.

the IHM per dataset type (which can be consulted in A), we have observed that:

- The growth in the number of observations is most frequently identified by the measures CLD, N2_{IHM} and $\text{Closeness}_{\text{IHM}}$. CLD is based on the probabilities of the features values given the class using a Naïve Bayes rule, N2_{IHM} is a neighborhood-based measure and $\text{Closeness}_{\text{IHM}}$ is also based on data similarity and neighborhood information.
- Although the growth in the number of features (higher dimensionality) was not captured by the classifiers, three measures identified such an increase of dimensionality, namely $\text{Degree}_{\text{IHM}}$, $\text{Closeness}_{\text{IHM}}$ and N2_{IHM}. In fact, the more attributes are considered, while the number of observations is kept fixed, data becomes more sparse and the density of points in the input space decreases, a feature that these IHM are able to capture.
- The increase in the number of classes is best identified by the measures kDN, DCP, CLD, N1_{IHM}, N2_{IHM} and H. While kDN, N1_{IHM}, N2_{IHM} and H are all neighborhood-based measures, DCP relies on a decision tree separation of the data and CLD is based on the Naïve Bayes rule.
- Class overlap is most often identified by the measures kDN, DCP, CLD, N1_{IHM}, N2_{IHM}, LSC_{IHM}, LSR, F1_{IHM}, U, $\text{Degree}_{\text{IHM}}$ and $\text{Closeness}_{\text{IHM}}$. Almost all measures were able to capture this complexity variation, which was reflected even in the ability of the input features to individually separate the classes (F1_{IHM}).
- The increase in the flip rate, representing the insertion of noise in the dataset, is most often identified by the measures kDN, DCP, CLD, N1_{IHM}, N2_{IHM}, LSC_{IHM}, LSR, U, $\text{Degree}_{\text{IHM}}$ and $\text{Closeness}_{\text{IHM}}$. They are almost the same IHM as those highlighted when class overlap was varied, except from the feature-based IHM.

4.2 Regression datasets

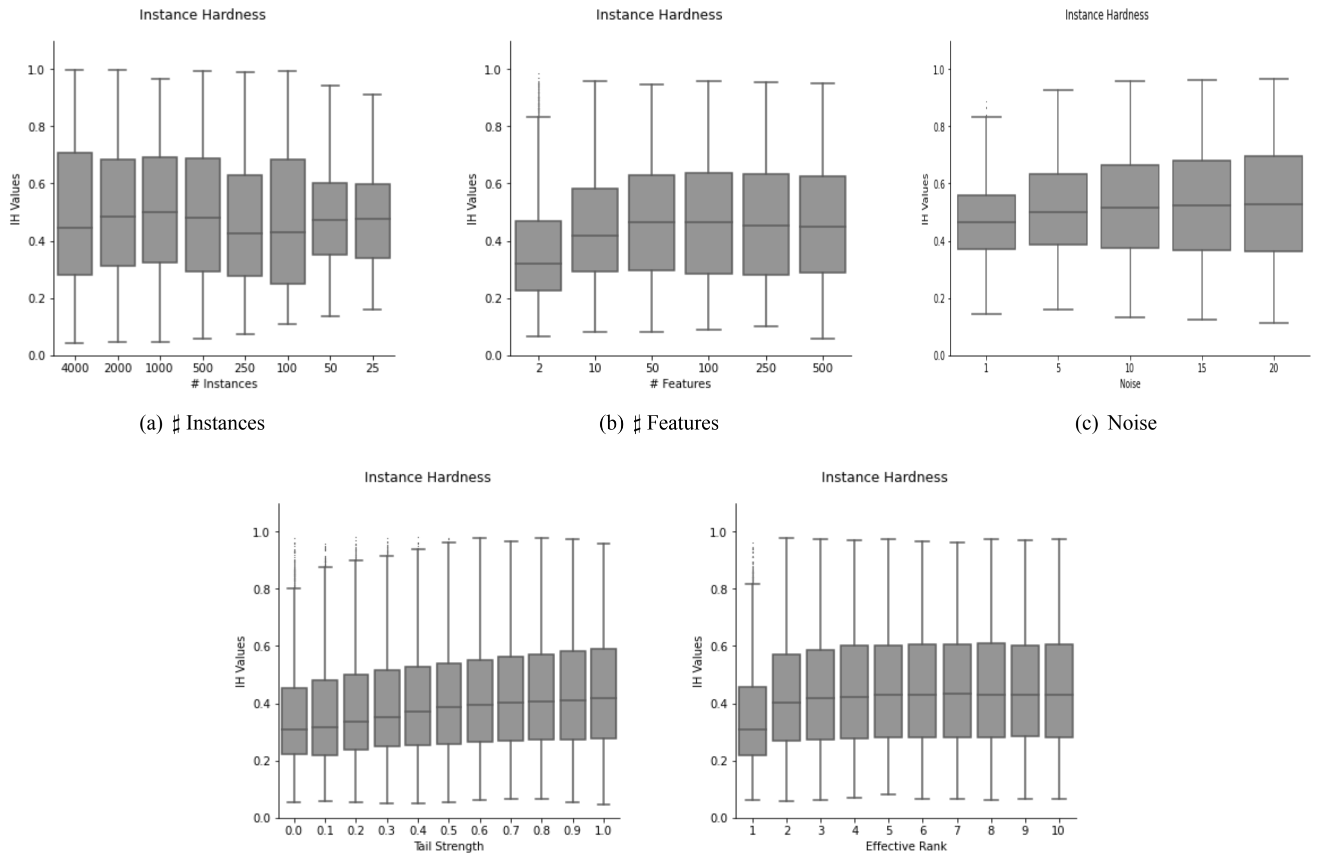
The regression datasets were generated using the `make_regression` package from the scikit-learn Python library [Pedregosa *et al.*, 2011]. It generates datasets with a linear relationship between the input features and the output values, where some types of noise can be added. The different characteristics which may affect the difficulty in solving a regression problem varied in this work are:

1. **Number of instances**: datasets with two informative input features and no noise, varying the number of instances as: 25, 50, 100, 250, 500, 1000, 2000 and 4000. The greater the number of instances for the same number of input features, the lower the difficulty level expected.
2. **Number of features**: datasets with 1000 instances and no noise, varying the number of features as: 2, 10, 50, 100, 250 and 500. Lesser features were tested in the case of regression problems due to an increased computational cost for higher dimensions. The greater the number of input features for the same number of instances, the greater the expected difficulty level, as data becomes more sparse in the input space.
3. **Noise**: datasets with two informative input features and 1000 instances, varying the standard deviation of a Gaussian noise applied to the output with the levels: 1, 5, 10, 15 and 20. The higher the noise level, the more difficult to solve the regression problem it is.
4. **Tail strength**: datasets with two informative input features, 1000 instances and no noise, but varying the values for tail strength from 0 to 1.0 with steps of 0.1. Increasing the tail strength makes the regression problem more difficult, as data becomes bad conditioned and the tail can be considered a noisy part of the data too.
5. **Effective rank**: datasets with two informative input features, 1000 instances and no noise, varying values for effective rank from 1 to 10 with steps of 1. The larger the effective rank value, the more correlated are the input features, which tends to affect negatively the regression results.

The same procedures adopted for the classification datasets are repeated here. First we compute the instance hardness levels as measured by Equation 2 for the regression datasets. The pool $\mathcal{L}$ of regression techniques used in Equation 2 was: AdaBoost, ν-SVM, RF, Extremely Randomized Trees, Regression Tree, GN, MLP, Bagging, Bayesian Automatic Relevance Determination, Kernel Ridge Regression, Stochastic Gradient Descent Regression and Passive-Agressive Regression. All regression techniques were run in a 5-fold cross-validation procedure and had hyperparameters tuned by an inner 3-fold cross-validation on the training folds.

The boxplots of the IH values for the datasets with different difficulty levels are shown in Figure 4, with the IH values in the y axis and the increasing levels of difficult in the x axis (in the case of the variation of number of instances, the values in the x axis are shown in decreasing order). As we can see in the plot, the IH values follow an increasing trend in most of the cases, except for when the number of instances is varied. In the last case, the results are inconsistent and there is no clear trend. The characteristics which seem to impact more the results of the regressors is an increased amount of noise in the labels and an increase in the tail strength. Both corresponds to the introduction of some type of noise in the regression problem.

The boxplots of some of the IHM are presented in Figure 5 (complete plots are presented in B). All of them follow the increasing trends of the different difficulty levels being considered. In Figure 5a, the smaller the size of the dataset concerning the number of instances, the S1_{IHM} values tend to increase. This happens because the dataset becomes less dense for lesser instances, making the outputs or neighbor instances more distant. As the number of input features increases, so does the S2_{IHM} values (Figure 5b). That is, for more features, the average distance between instances with neighboring outputs tends to be larger than that computed for lesser features. This is an effect of a higher sparsivity of the input data when more features are added, while the number of instances remains fixed. By introducing more noise into the output labels of the instances in the datasets (Figure 5c), the S3_{IHM} also increases, which corresponds to the Squared Error (SE) of a nearest neighbor regressor prediction. In fact, the



(a) ♯ Instances

(b) ♯ Features

(c) Noise

(d) Tail Strength

(e) Effective Rank

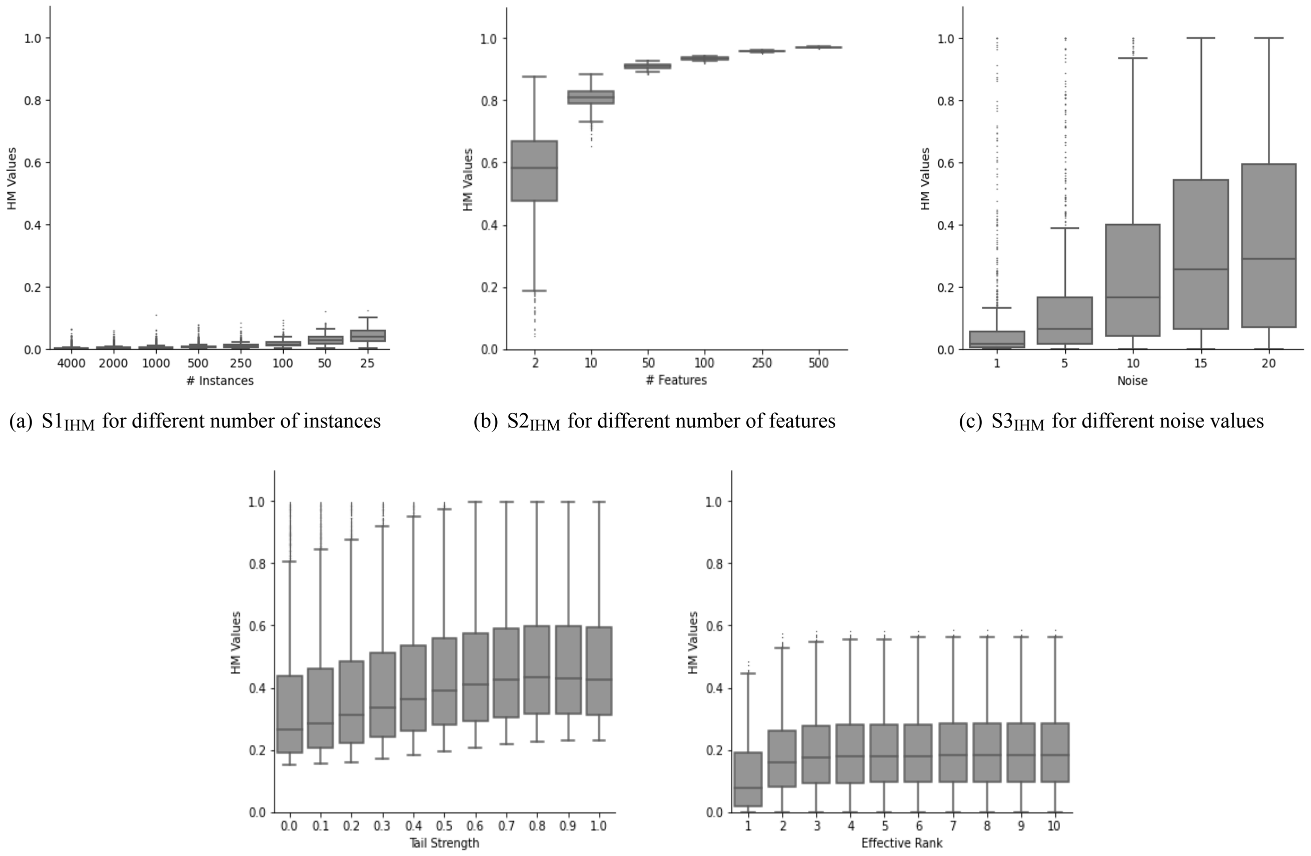
Figure 4. IH values measured at regression datasets of increasing complexities according to different difficulty sources.

average predictions given by the nearest neighbor rule will be influenced by the perturbation of the output values. Varying the tail strength influences in the density of the proximity graphs built from data, which is reflected in the Degree$_{\text{IHM}}$ values, as shown in Figure 5d. Finally, the LE values tend to increase for higher effective rank values (Figure 5e). This variation concerns the input features, whose correlation affects the results of the linear regressor employed in LE computation.

Figure 6 shows a heatmap of the Spearman correlation between the IHM and the IH as measured by Equation 2. The redder the color, the higher the correlation. We can notice that, in the case of regression dataset, the correlations are usually lower than those observed for the classification datasets. There are some specific combinations of IHM and sources of difficulty that present highlighted results. For instance, S1$_{\text{IHM}}$ and S2$_{\text{IHM}}$ are very correlated to the increase in the number of features, while noise was better captured by LE. CFE did not present much variations and was not so effective in our scenarios. But one must notice that in almost all datasets generated, the number of input features is only two, which may have affected the results. TD again had unstable results in our tested scenarios.

Looking at the individual plots from the measures (presented in Appendix B), we can highlight the following trends:

- The growth in the number of instances did not affect the predictive results of the regressors uniformly. But IHM such as S1$_{\text{IHM}}$, TD and Degree$_{\text{IHM}}$ were able to reveal the variation in the number of instances, that is, these measures increase uniformly as the number of instances decreases. S1$_{\text{IHM}}$ and Degree$_{\text{IHM}}$ are both based on neighborhood information, while TD is based on partitioning the input space with regression trees. In contrast, the dynamics of the IH measured by the regressors when the number of instances is varied seems to be better captured by measures S2$_{\text{IHM}}$ (neighborhood-based) and LE (based on linear fit).
- Again, the predictive results of the regressors were not affected uniformly for increased numbers of input features. But the IHM S1$_{\text{IHM}}$, S2$_{\text{IHM}}$ and Degree$_{\text{IHM}}$ were effective to detect the dimensionality increase imposed by varying the number of features, that is, these measures increase uniformly as the number of input features increases. A higher dimension implies in more sparsity in the dataset, which these neighborhood distance-based measures are able to capture. Here, the dynamics of the IH measured by the regressors when the number of input features is varied seems to be better captured by the neighborhood-based measure S3$_{\text{IHM}}$.
- Varying the noise levels in the labels was better captured by the IHM LE and S3$_{\text{IHM}}$. They both consider the error of some simple regressors, being a linear regressor for LE and a nearest neighbor regressor for S3$_{\text{IHM}}$.
- Another source of noise is input by varying the tail strength parameter of the data generator, whose increase



(a) $S1_{IHM}$ for different number of instances

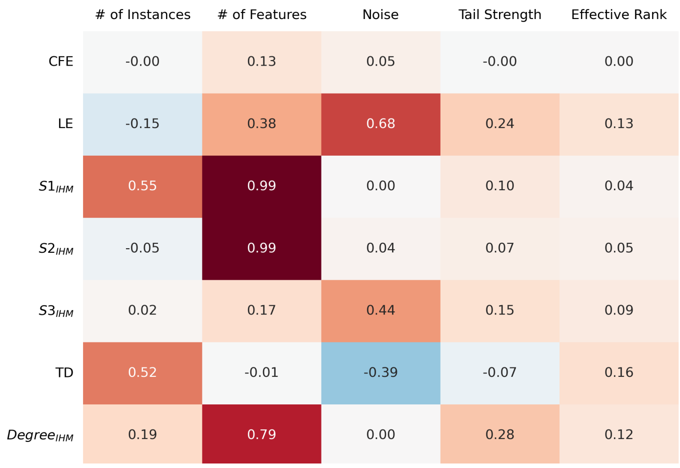
(b) $S2_{IHM}$ for different number of features

(c) $S3_{IHM}$ for different noise values

(d) $Degree_{IHM}$ for different tail strength values

(e) LE for different effective rank values

Figure 5. IHM values measured at datasets of increasing complexities according to different difficulty sources for regression problems.



	# of Instances	# of Features	Noise	Tail Strength	Effective Rank
CFE	-0.00	0.13	0.05	-0.00	0.00
LE	-0.15	0.38	0.68	0.24	0.13
$S1_{IHM}$	0.55	0.99	0.00	0.10	0.04
$S2_{IHM}$	-0.05	0.99	0.04	0.07	0.05
$S3_{IHM}$	0.02	0.17	0.44	0.15	0.09
TD	0.52	-0.01	-0.39	-0.07	0.16
$Degree_{IHM}$	0.19	0.79	0.00	0.28	0.12

Figure 6. Correlations between IHM and instance hardness measured by multiple regressors for the different regression datasets generated.

was better reflected by IHM such as LE, $S2_{IHM}$, $S3_{IHM}$ and $Degree_{IHM}$. They are neighborhood-based IHM ($S2_{IHM}$, $S3_{IHM}$ and $Degree_{IHM}$) and a measure devoted to estimate deviations from a linear fit (LE).

- Increasing the correlation of the input features was better captured by the IHM LE, $S2_{IHM}$, $S3_{IHM}$ and $Degree_{IHM}$, which concern on linearity (LE) and neighborhood information ($S2_{IHM}$, $S3_{IHM}$ and $Degree_{IHM}$).

To verify the effect of using more input features, we have generated the same regression datasets but fixed the number of input features in five (except for the scenario where the number of features is varied, which remains the same). The heatmap of this test is presented in B. Overall, the correlations remained similar for the scenarios of varying the number of instances and noise levels. For tail strength and effective rank, the correlation increased for most IHMs. The $S1_{IHM}$ measure was the one that most correlated to the hardness measured by the multiple regressors, except for noise variation.

5 Conclusions

This paper investigated different measures for estimating how hard it is to predict the labels of individual instances of a classification or regression dataset in ML, named *instance hardness measures*. They present different perspectives on why an instance is more difficult to predict than another in a dataset. Instance hardness can also be assessed by the predictive performance of multiple ML models for a given instance and is higher when their predictions consistently differ from the label registered in the dataset.

By generating synthetic datasets, we showed experimentally that each IHM can be more effective in reflecting the increase in the difficulty level of the instances when different sources of complexity are concerned. In the case of classification datasets, usually the overlap of the classes is captured more effectively and by more IHM. For regression datasets, varying the number of input features in the data has influenced more the instance hardness values. In both cases, there is a prominence of IHM based on neighborhood information, which tend to be more effective in instance hardness analysis.

As future work, it is also important to validate the use of the IHM in real datasets and try to identify different types of problematic instances more effectively. Finally, validating the usage of the IHM in applications such as data preprocessing, curriculum learning and active learning are research paths worth future investigations.

Acknowledgements

The authors would like to thank the financial support of FAPESP under grant number 2021/06870-3.

Funding

This research was funded by FAPESP and CNPq.

Authors' Contributions

ACL contributed to the conception of this study. GT, VSN and PYAP implemented the codes and performed the experiments. ACL is the main contributor and writer of this manuscript. All authors read and approved the final manuscript.

Competing interests

The authors declare that they do not have any competing interests.

Availability of data and materials

The datasets generated and/or analysed during the current study are available under request.

References

Arruda, J. L., Prudêncio, R. B., and Lorena, A. C. (2020). Measuring instance hardness using data complexity measures. In *Brazilian Conference on Intelligent Systems*, pages 483–497. Springer.

Cruz, R. M., Sabourin, R., and Cavalcanti, G. D. (2018). Dynamic classifier selection: Recent advances and perspectives. *Information Fusion*, 41:195–216.

Cruz, R. M., Sabourin, R., Cavalcanti, G. D., and Ren, T. I. (2015). Meta-des: A dynamic ensemble selection framework using meta-learning. *Pattern recognition*, 48(5):1925–1935.

Garcia, L. P., de Carvalho, A. C., and Lorena, A. C. (2015). Effect of label noise in the complexity of classification problems. *Neurocomputing*, 160:108–119.

Leyva, E., González, A., and Pérez, R. (2014). A set of complexity measures designed for applying meta-learning to instance selection. *IEEE Transactions on Knowledge and Data Engineering*, 27(2):354–367.

Leyva, E., González, A., and Pérez, R. (2015). Three new instance selection methods based on local sets: A comparative study with several approaches from a bi-objective perspective. *Pattern Recognition*, 48(4):1523 – 1537.

Lorena, A. C., De Carvalho, A. C., and Gama, J. M. (2008). A review on the combination of binary classifiers in multiclass problems. *Artificial Intelligence Review*, 30:19–37.

Lorena, A. C., Garcia, L. P., Lehmann, J., Souto, M. C., and Ho, T. K. (2019). How complex is your classification problem? a survey on measuring classification complexity. *ACM Computing Surveys (CSUR)*, 52(5):1–34.

Lorena, A. C., Maciel, A. I., de Miranda, P. B., Costa, I. G., and Prudêncio, R. B. (2018). Data complexity meta-features for regression problems. *Machine Learning*, 107(1):209–246.

Martínez-Plumed, F., Prudêncio, R. B., Martínez-Usó, A., and Hernández-Orallo, J. (2019). Item response theory in ai: Analysing machine learning classifiers at the instance level. *Artificial intelligence*, 271:18–42.

Moraes, J. V., Reinaldo, J. T., Ferreira-Junior, M., Silva Filho, T., and Prudêncio, R. B. (2022). Evaluating regression algorithms at the instance level using item response theory. *Knowledge-Based Systems*, 240:108076.

Morais, G. and Prati, R. C. (2013). Complex network measures for data set characterization. In *2013 Brazilian Conference on Intelligent Systems*, pages 12–18. IEEE.

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.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., *et al.* (2011). Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830.

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

Schweighofer, E. (2021). Data-centric machine learning: Improving model performance and understanding through dataset analysis. In *Legal Knowledge and Information Systems: JURIX 2021*, volume 346, page 54. IOS Press.

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

Sowkarthika, B., Gyanchandani, M., Wadhvani, R., and Shukla, S. (2023). Data complexity-based dynamic ensembling of svms in classification. *Expert Systems with Applications*, 216:119437.

Torquette, G. P., Nunes, V. S., Paiva, P. Y., Neto, L. B., and Lorena, A. C. (2022). Characterizing instance hardness in classification and regression problems. *arXiv preprint arXiv:2212.01897, Proceedings of KDMile 2022*.

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

A Complete results for classification datasets

Figures 7 to 11 present the boxplots of all IHM for the classification datasets generated in this work. The last set of boxplots corresponds to the IH as measured by a collection of classification algorithms (Equation 1), while the others are the IHM presented in Section 3.1.

B Complete results for regression datasets

Figures 12 to 16 present the boxplots of all IHM for the regression datasets generated in this work. The last set of boxplots corresponds to the IH as measured by a collection of regression algorithms (Equation 2), while the others are the IHM presented in Section 3.2. The heatmap between the IHM and the instance hardness measured by the regressors for datasets with five dimensions (except for the ♯Features, which varies the number of input features) is presented in Figure 17.

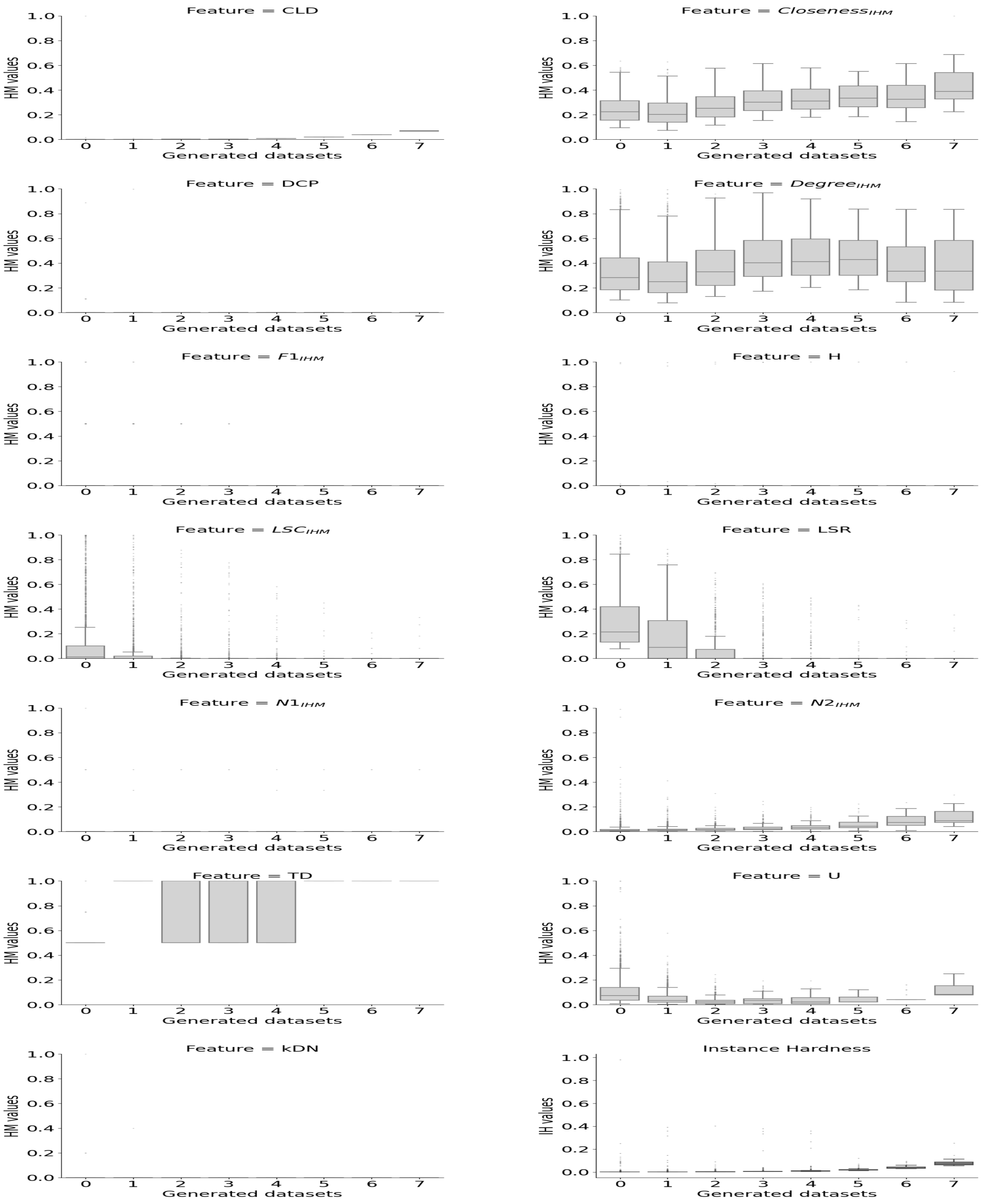


Figure 7. IH and IHM when varying number of instances in each classification dataset (ordered from largest to smallest dataset).

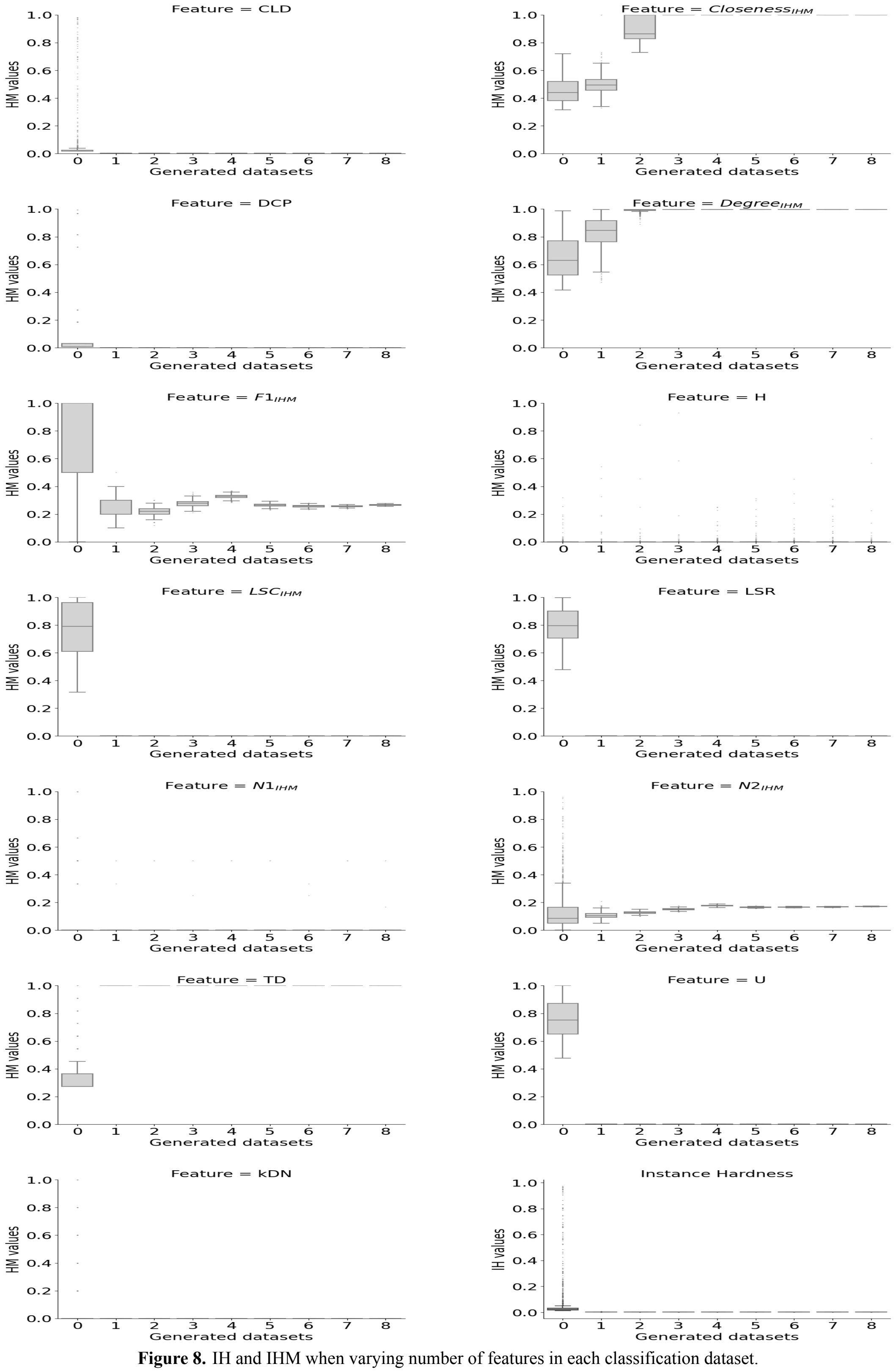


Figure 8. IH and IHM when varying number of features in each classification dataset.

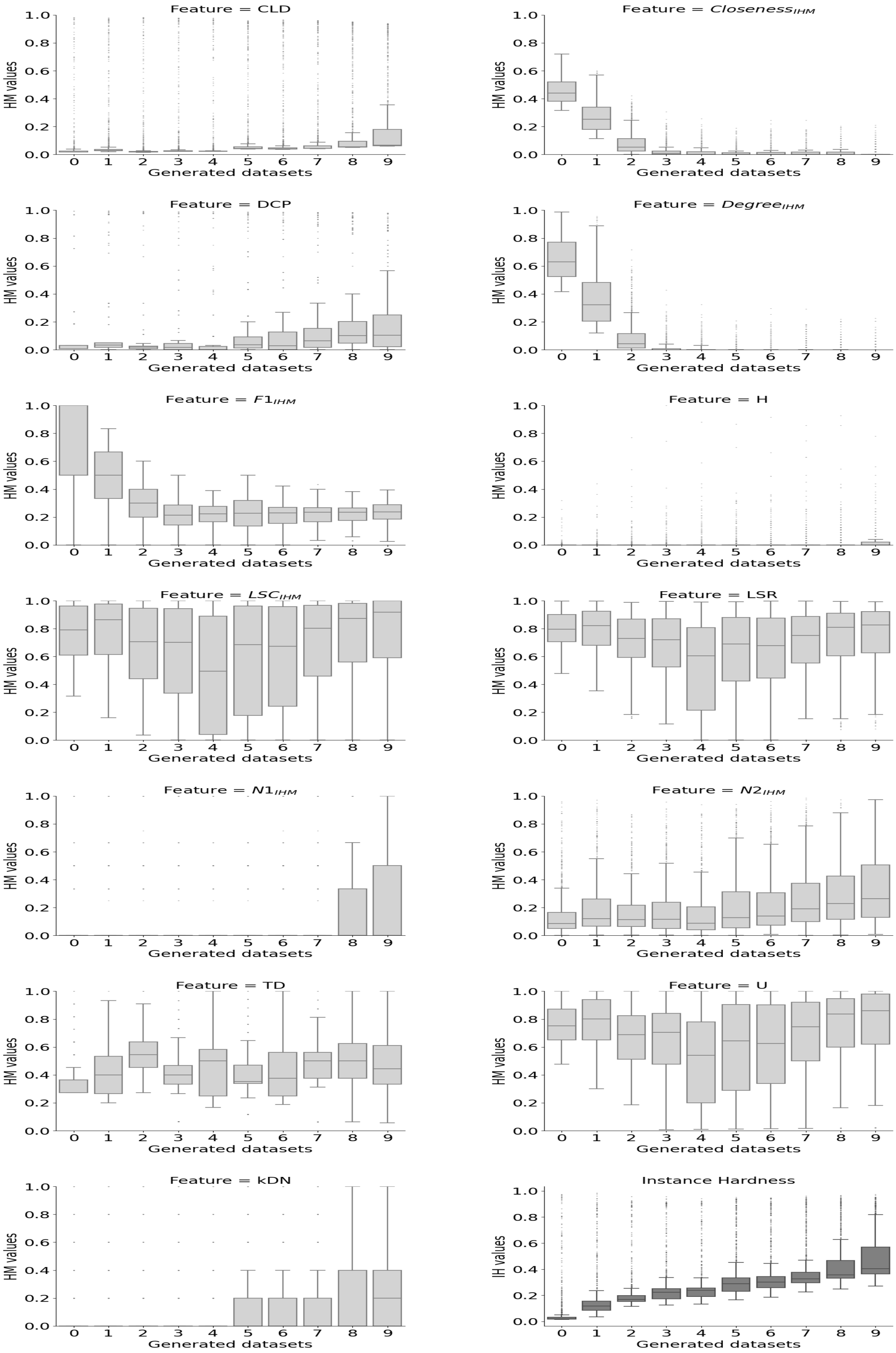


Figure 9. IH and IHM when varying number of classes in classification each dataset.

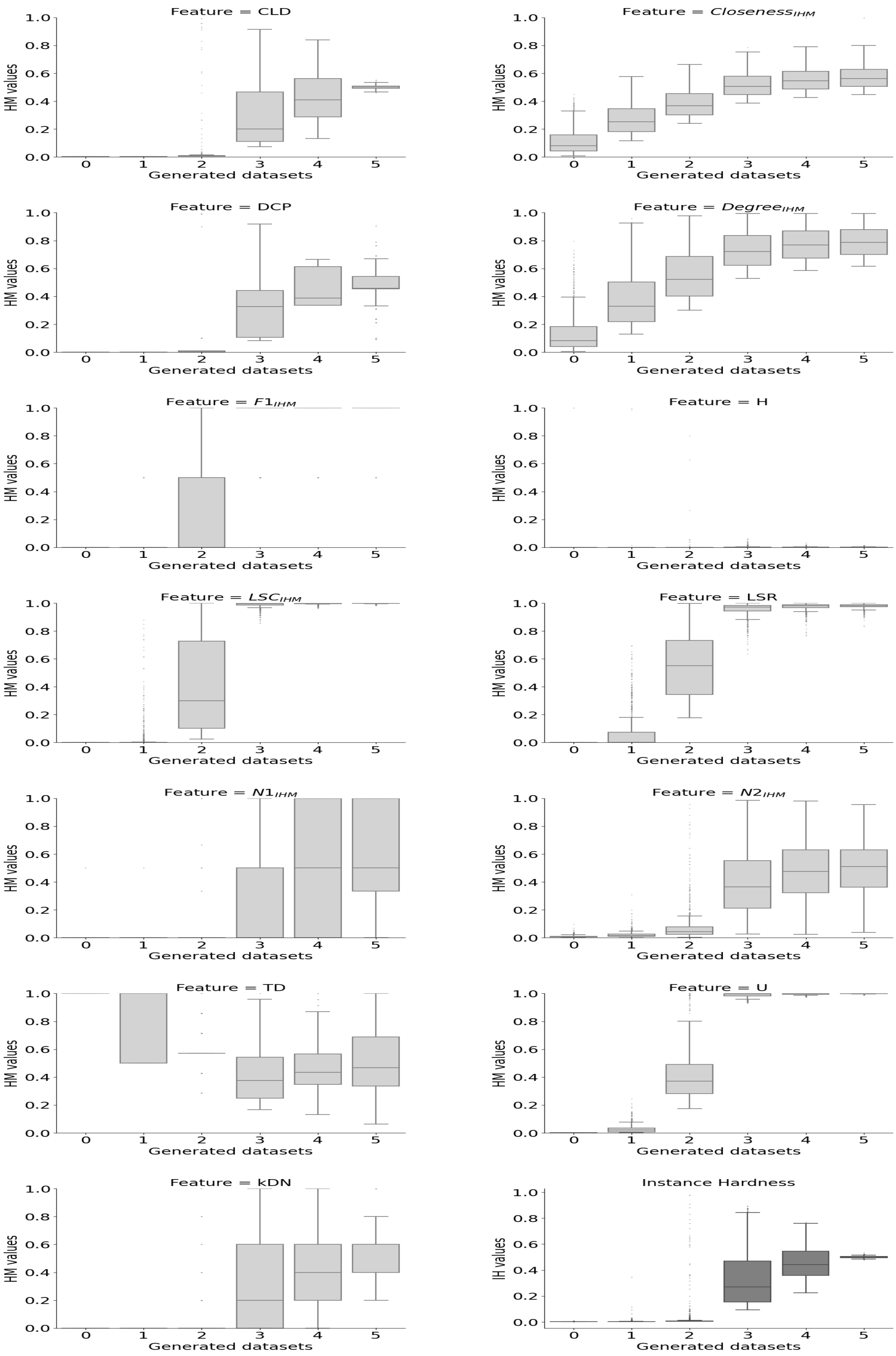


Figure 10. IH and IHM when varying overlap of the classes in each classification dataset.

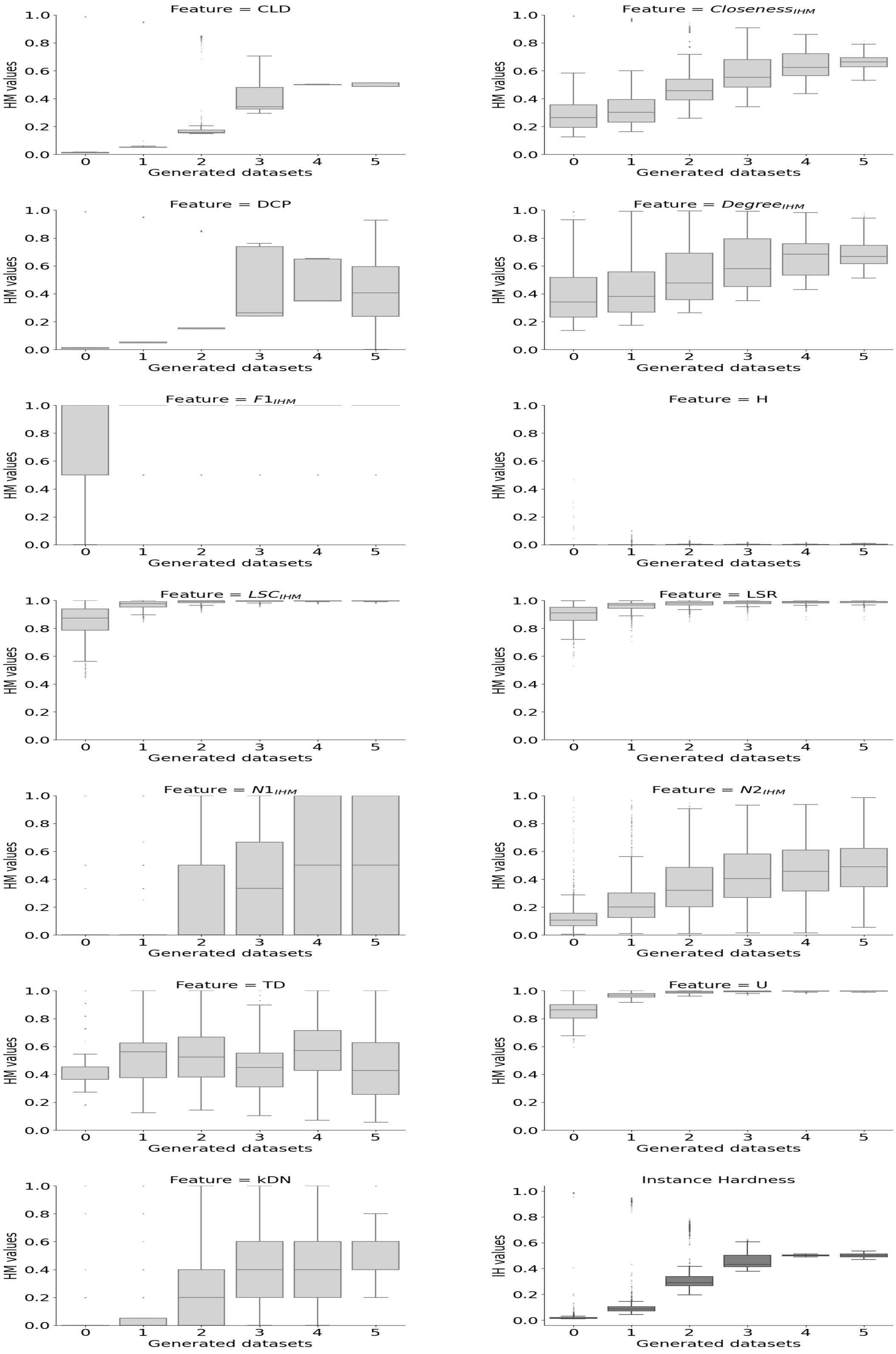


Figure 11. IH and IHM when varying label flip in each classification dataset.

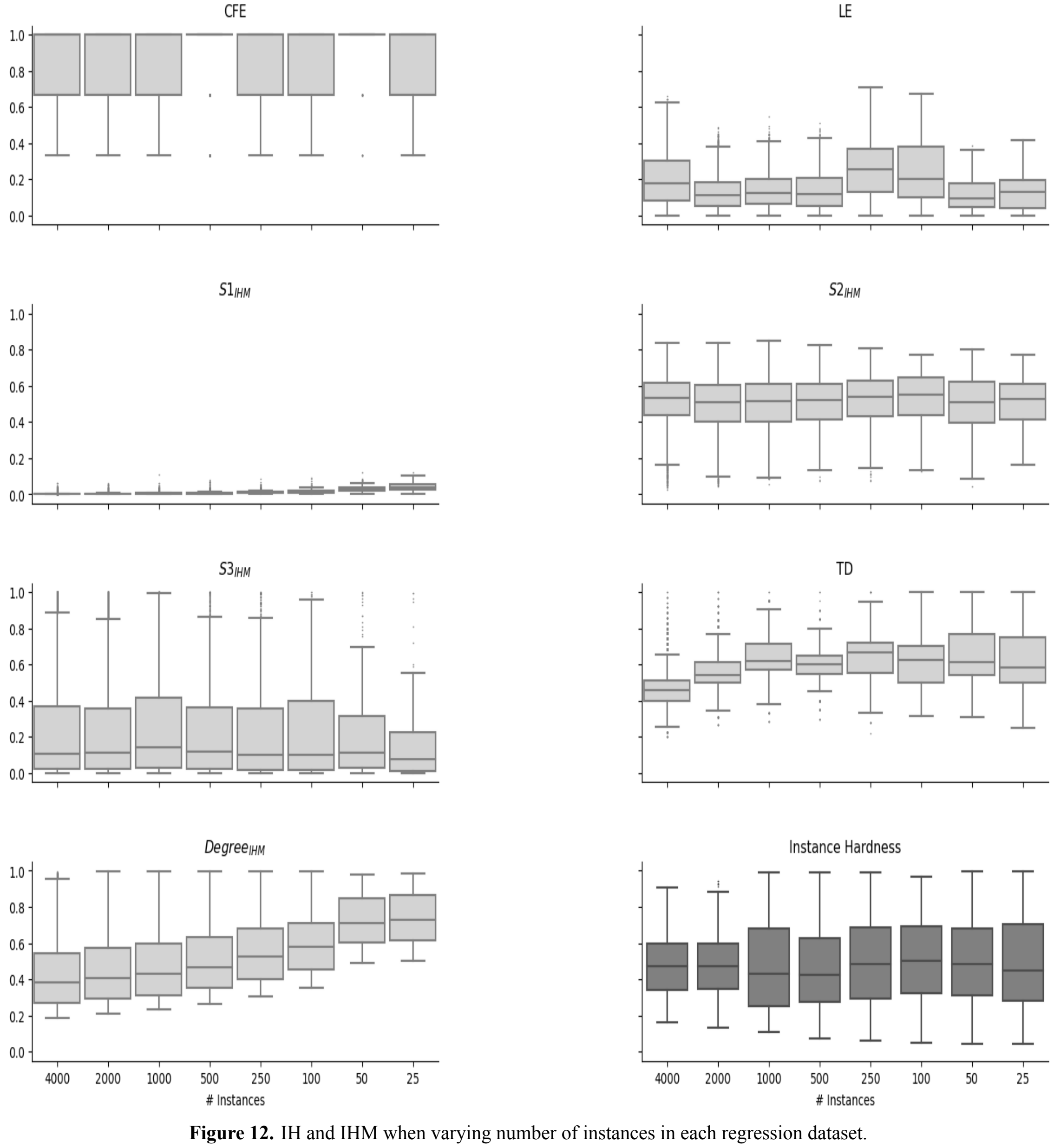


Figure 12. IH and IHM when varying number of instances in each regression dataset.

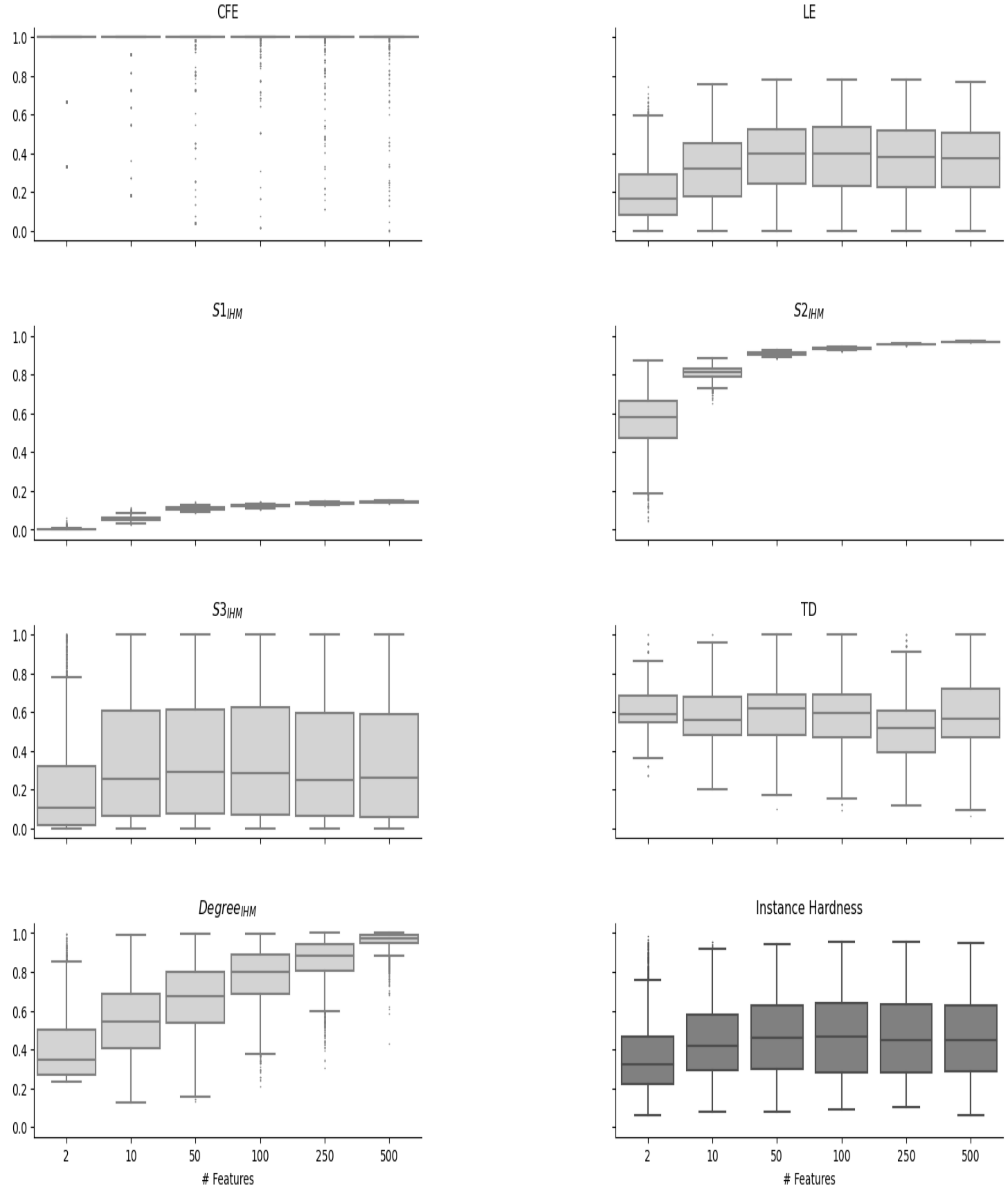


Figure 13. IH and IHM when varying number of features in each regression dataset.

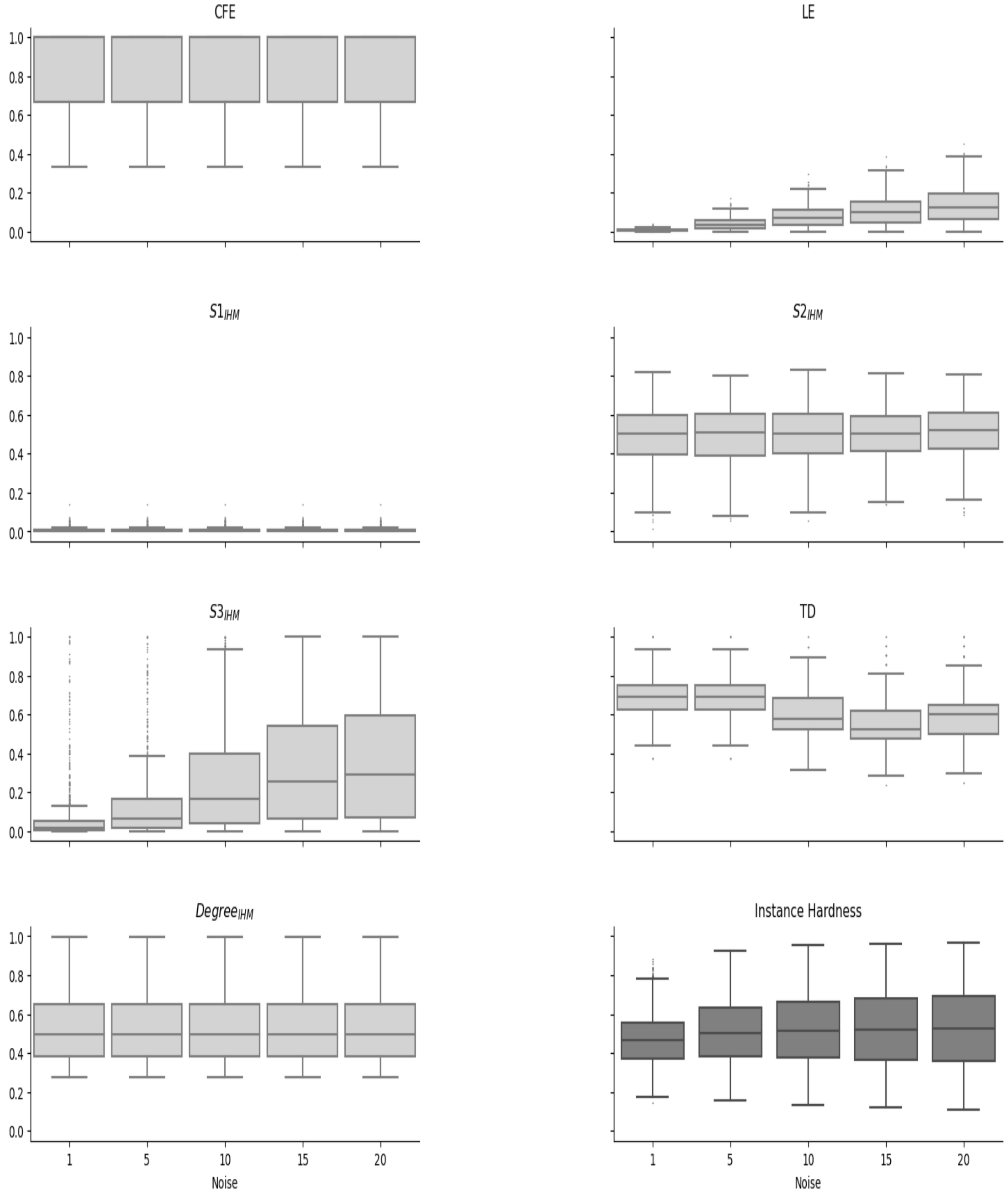


Figure 14. IH and IHM when varying noise in each regression dataset.

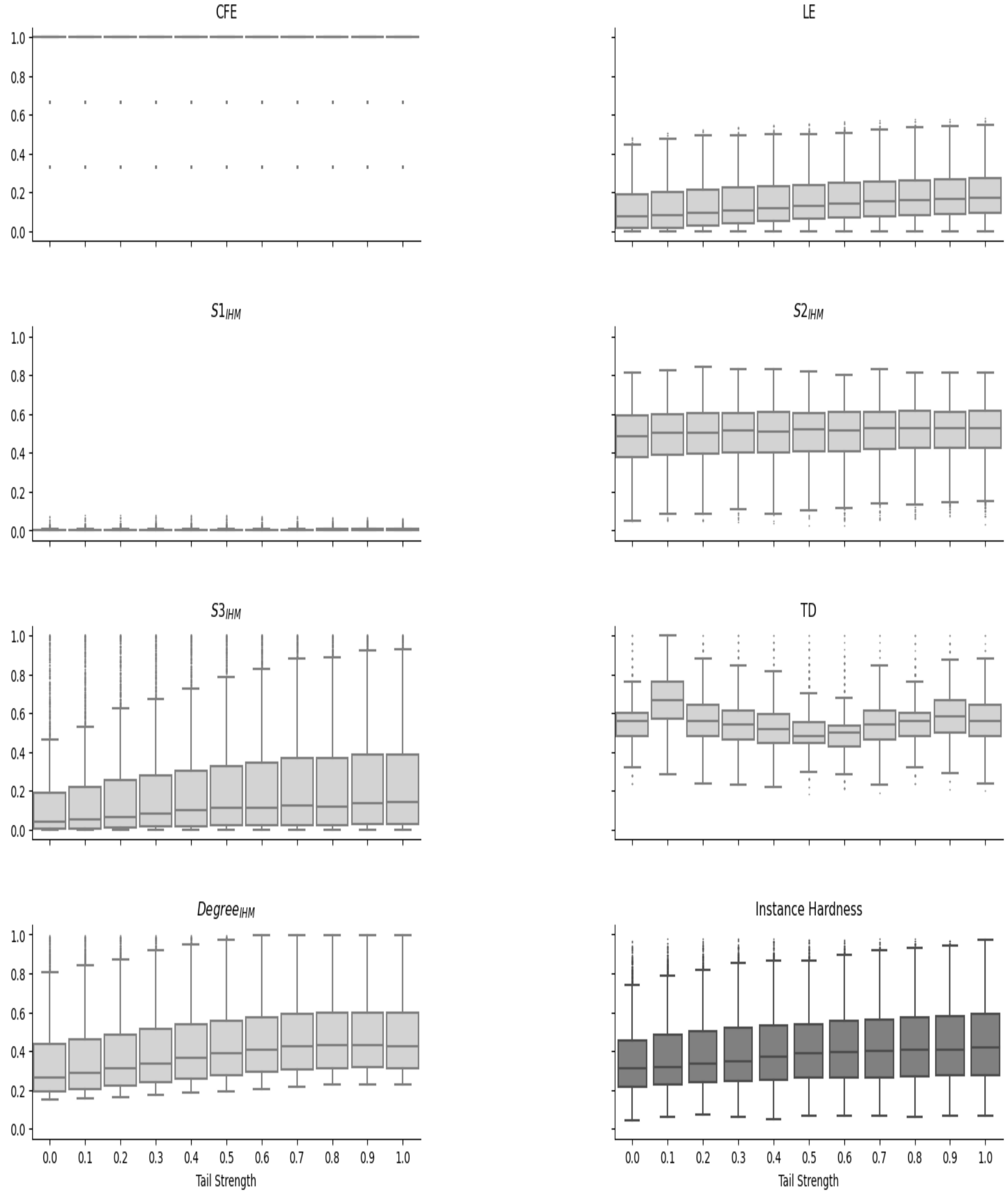


Figure 15. IH and IHM when varying tail strenght values in each regression dataset.

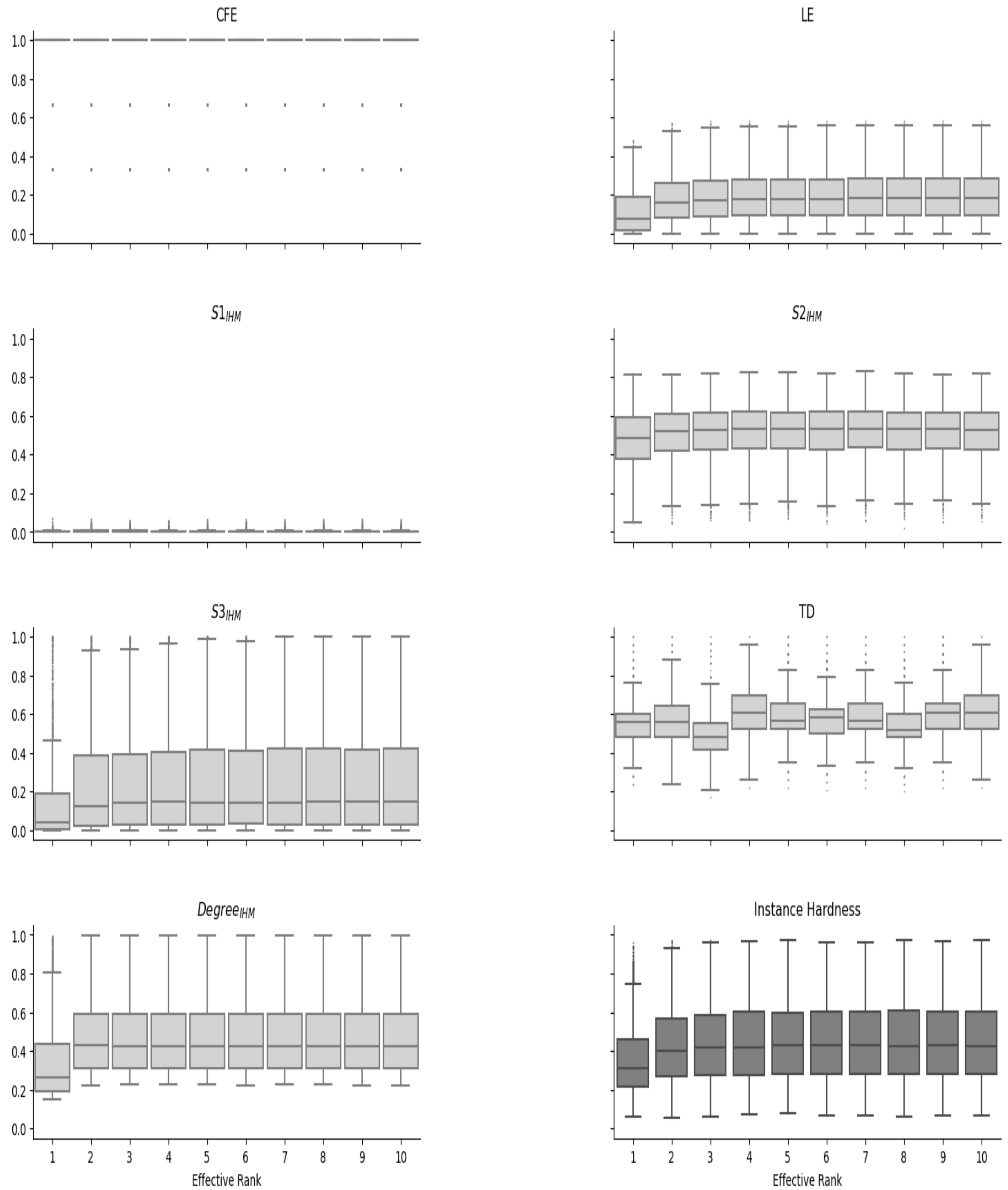
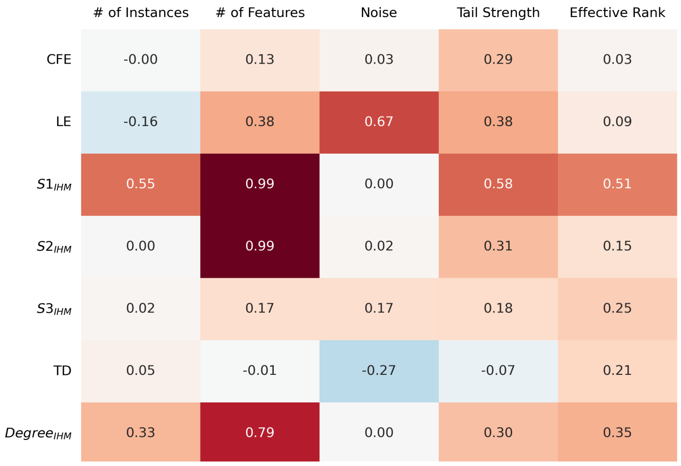


Figure 16. IH and IHM when varying effective strength values in each regression dataset.



	# of Instances	# of Features	Noise	Tail Strength	Effective Rank
CFE	-0.00	0.13	0.03	0.29	0.03
LE	-0.16	0.38	0.67	0.38	0.09
$S1_{IHM}$	0.55	0.99	0.00	0.58	0.51
$S2_{IHM}$	0.00	0.99	0.02	0.31	0.15
$S3_{IHM}$	0.02	0.17	0.17	0.18	0.25
TD	0.05	-0.01	-0.27	-0.07	0.21
$Degree_{IHM}$	0.33	0.79	0.00	0.30	0.35

Figure 17. Correlations between IHM and instance hardness measured by multiple regressors for different regression datasets. Five dimensions are used in all scenarios, except for the ♯ Features, which varied more in the number of features.