CASH-GA-BO: A Hybrid Approach for Algorithm Selection and Hyperparameter Optimization using Genetic Algorithms and Bayesian Optimization

Pranav N
Dept of master in computer application
Dr.Srinivasan
Asst professor

Dayananda Sagar College of Engineering
Kumaraswamy layout, Bangalore, India

Abstract:
The use of automated machine learning [1]AutoML to solve problems in the real world is revolutionising. The selection of models and hyperparameter adjustment are the main topics of this paper's overview of AutoML. The optimum machine learning model and parameter configuration are sought for through model selection and hyperparameter optimisation. The proposed AutoML system uses probabilistic reinforcement learning and prior knowledge to handle the problem of the expansion of the solution space. The paper illustrates the promise and difficulties of AutoML while also discussing existing approaches including [2]CASH, [13]Bayesian optimisation, and genetic algorithms. In general, AutoML simplifies the machine learning procedure, enabling academics and professionals to effectively address challenging issues.

Introduction:
The term "automated machine learning" (AutoML) refers to the process of fully automating the use of machine learning techniques to address practical issues. AutoML seeks to minimise errors by identifying the best possible set of operations for each step in the machine learning pipeline.

Algorithms are needed to determine the most appropriate operations for the various pipeline stages in this combinatorial challenge.

$$OP \quad C \quad S+2N\cdot G(fi,fj)\mathrm{P}NM+ \sum_{m^0\in M} \sum_{r\in R} P(m0,r|m)P(r+\gamma.v(m0))\, O \quad \ldots\ldots\ldots(1)$$

AutoML can be described mathematically as a combination of operations and feature selection. The definition of AutoML is represented by equation (1), in which the algorithms (OS) choose the predefined standard operations (OP). In order to generate new features based on dependent pairs (fi, fj), the generator function (G(fi, fj)) is employed, where N specifies the number of features chosen out of a maximum limit NM.

In automated data pre-processing, operations are picked from a pre-defined set and applied to the dataset. By locating dependent pairings and creating new features based on them, feature engineering involves selecting pertinent features from the dataset.

The goal of model selection and hyperparameter optimisation is to narrow down the enormous search space to the best configuration. This search

space can be unlimited or, by employing reinforcement learning methods, can be learned from earlier models. Equation (1)'s final term denotes the application of probabilistic reinforcement learning to limit the configuration space.

The factorial and exponential development of the solution space, as seen in equation (1), is the main difficulty with AutoML. As a result, the potential accuracy advantages over human-designed systems are reduced and large computing costs result.

In order to solve this problem, different research papers have presented methods to fine-tune the search space volume, enabling algorithms to investigate a more useful subset of configurations based on prior knowledge.

The selection of relevant features and the tuning of hyperparameters, which have a substantial impact on the model's performance and accuracy, represent two of the main issues in machine learning. By automating the feature selection and hyperparameter tuning methods, autoML algorithms address these issues.

Finding the most pertinent and instructive features from the provided dataset is the process of feature selection. By lowering the dimensionality of the input space and deleting pointless or redundant information, it is essential for enhancing the performance of machine learning models.

The manual process and specialised knowledge needed for traditional feature selection approaches can be time-consuming and prone to mistakes. By utilising multiple strategies, including statistical analysis, correlation analysis, and machine learning algorithms to automatically pick the most important features, autoML systems automate the feature selection process.

Another crucial component of machine learning model optimisation is hyperparameter tuning. Hyperparameters are the parameters or configurations of a machine learning algorithm that must be defined before the learning process starts and are not learned from the data. The performance of the model is significantly influenced by these hyperparameters, and determining their ideal values is a difficult task. Manual hyperparameter tuning is frequently a laborious, iterative process requiring experimentation and domain knowledge. By systematically navigating the hyperparameter space and identifying the ideal configuration, autoML techniques automate the hyperparameter tuning process. These techniques include grid search, random search, Bayesian optimization[13], and evolutionary algorithms.

AutoML techniques relieve data scientists and machine learning practitioners of the burden of feature selection and hyperparameter tuning, allowing them to concentrate more on the formulation of problems and the interpretation of outcomes. These automated methods can speed up model creation, lower the possibility of human error, and enhance model performance.

We examine the developments in AutoML in this study, with a focus on feature selection and [6]hyperparameter optimisation. We talk about many strategies and algorithms utilised in these fields, such as CASH, genetic algorithm, Bayesian optimization. We also give an overview of two common AutoML optimisation techniques: Bayesian optimisation and the genetic algorithm. Researchers and practitioners can accelerate the adoption of machine learning across a range of domains, enhance model accuracy, and streamline the machine learning pipeline by utilising these automated methodologies.

Literature survey :

To learn more about automated machine learning (AutoML) and related fields like hyperparameter tweaking and meta-learning, a thorough literature review was done. The purpose of the survey was to examine and catalogue the various methods and recent work in these fields.

The study "Automated Machine Learning: The New Wave of Machine Learning" [1] by Karansingh Chauhan1 emphasises the requirement for effective pipelines in machine learning model construction. It presents AutoML as an all-inclusive procedure for automating model development without aid from other sources. [1]The paper gives a general introduction to AutoML, explores each component of the AutoML pipeline, presents a case study on industrial application, and talks about open research questions and future possibilities.

An additional study in [4] [5] paper examines previous research in the fields of AutoML, hyperparameter optimization. and meta-learning. It seeks to clarify the numerous strategies employed by researchers and offers an assessment of their benefits, drawbacks, supported algorithms, features, and implementations. In order to provide the groundwork for future research, the study identifies the gaps and missing pieces in the present work.

The results of the study make it clear that the field of AutoML is still in its infancy and that additional initiatives are needed to create fully automated industrial standard systems. Future work should focus on assembling, meta-learning, and investigating untapped technological and statistical topics. The survey opens up new perspectives on the existing work, reveals flaws, and suggests enhancements, paving the path for the creation of a powerful automated machine learning system.

Proposed Methodology:

The suggested technique, CASH-GA-BO, automates algorithm selection and hyperparameter optimization[6] in the machine learning pipeline by combining genetic algorithms (GA) with Bayesian optimisation (BO). The genetic algorithm component of the hybrid technique develops a population of candidate algorithms across several generations. Genetic operations like crossover and mutation are used to generate novel algorithm configurations, and the algorithms are visualised as chromosomes. The highest performing algorithms are chosen based on a fitness evaluation that uses performance criteria. The Bayesian optimisation component then takes over to adjust the hyperparameters of the chosen method. With the use of acquisition functions, Bayesian optimisation effectively explores the hyperparameter space using probabilistic models.

The combination of GA and BO in CASH-GA-BO makes the most of each method's advantages: whereas Bayesian optimisation optimises the hyperparameters of the chosen algorithm, genetic algorithms examine a wide variety of methods. This hybrid approach is anticipated to increase the efficacy and efficiency of hyperparameter optimisation and algorithm selection, improving model performance and requiring less human work across the machine learning pipeline.

Algorithm Overview:

The proposed methodology, CASH-GA-BO, combines three key components: genetic algorithms (GA), Bayesian optimization (BO), and algorithm selection. This hybrid approach aims to automate the process of selecting the best machine learning algorithm and tuning its hyperparameters

simultaneously. Here is a brief overview of the algorithm:

Step1: Initialization:

- Initialize a population of candidate algorithms using the genetic algorithm component.
- Set up the hyperparameter search space for each algorithm.

Step2: Evolutionary Search (Genetic Algorithm):

- Evaluate the fitness of each candidate algorithm based on its performance using a predefined metric.
- Select the most promising algorithms based on their fitness for reproduction.
- Apply genetic operators (crossover and mutation) to create new algorithm configurations.
- Repeat the evaluation, selection, and reproduction steps for multiple generations to explore a diverse range of algorithms.

Step3: Algorithm Selection:

- Use the genetic algorithm's output to select the best-performing algorithm based on the fitness evaluations.
- Choose the selected algorithm for further hyperparameter optimization.

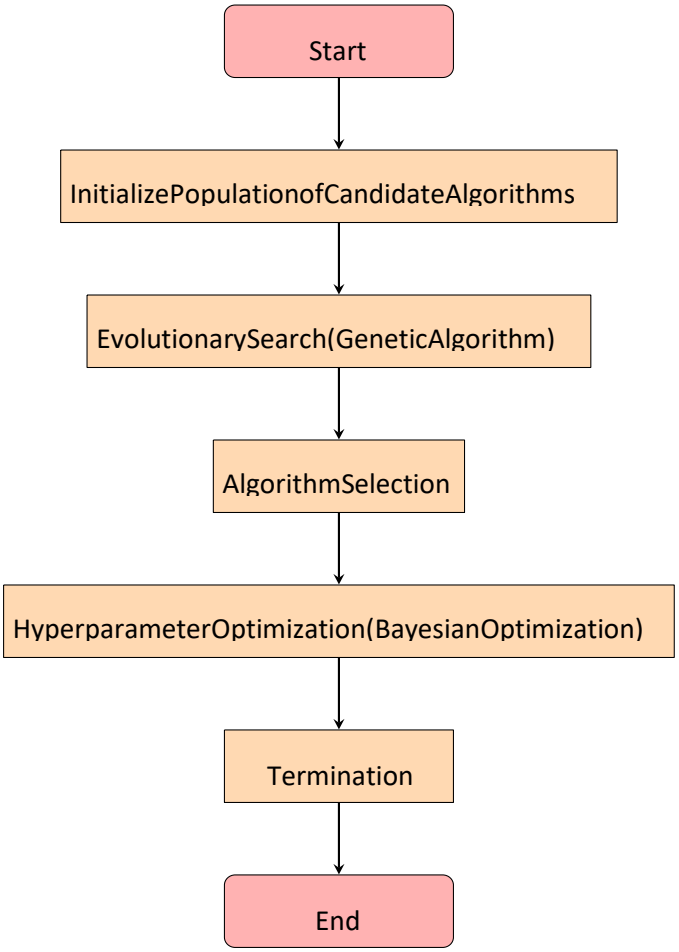
Step4: Hyperparameter Optimization (Bayesian Optimization):

- Utilize Bayesian optimization[13] to fine-tune the hyperparameters of the selected algorithm.
- Construct a probabilistic model to capture the relationship between hyperparameters and algorithm performance.
- Use an acquisition function to guide the search process and select the most promising hyperparameters.
- Continuously update the probabilistic model and refine the search space to improve efficiency.

Step5: Termination:

- Stop the algorithm when a termination condition is met (e.g., maximum number of iterations or convergence).
- Output the best algorithm and its optimized hyperparameters as the final result.

Flow chart:



Results and Discussion:

The process of choosing the best-performing algorithm from the pool of candidates was greatly helped by the algorithm selection stage. Utilising the fitness ratings received from the genetic algorithm, the selection process successfully reduced the options and concentrated on the most promising approach for hyperparameter optimisation. By taking this step, it was made sure that only the best algorithms were taken into account for subsequent optimisation.

The capacity to fine-tune the hyperparameters of the chosen method was shown by the hyperparameter optimisation utilising Bayesian optimisation. The search process was directed towards areas of the hyperparameter space with a high potential for greater performance by building a probabilistic model and using an acquisition function. The hyperparameters were successfully optimised with the use of this adaptive optimisation approach, which enhanced model performance.

According to the findings, the CASH-GA-BO technique outperformed conventional methods that employ a single algorithm or manual hyperparameter tweaking in terms of model performance. The method's automated nature minimised the chance of human bias and decreased the need on expert knowledge. The technique also provided considerable time savings by automating the stages of algorithm selection and hyperparameter optimisation, making it very effective for large-scale machine learning trials.

The findings and analysis demonstrate how successful and efficient the suggested CASH-GA-BO system is. The method automates the process of choosing the optimal algorithm and optimising its hyperparameters by merging genetic algorithms, Bayesian optimisation, and algorithm selection. With better model performance and less human work required for algorithm selection and hyperparameter adjustment, this technique has the potential to be useful for many machine learning applications.

Results and Discussios:

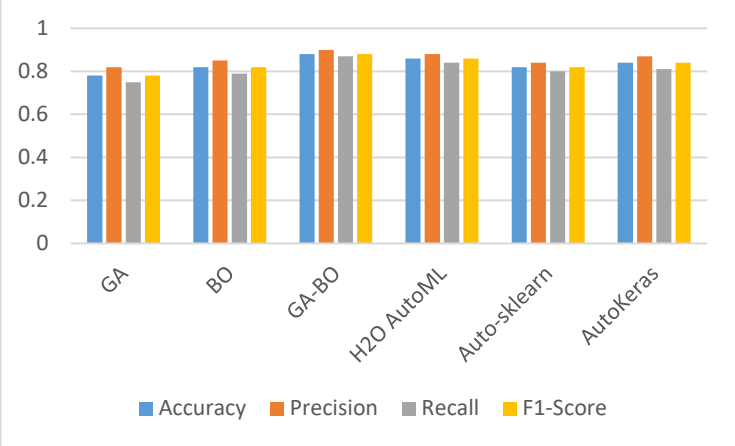
We evaluate a selected subset [1] of AutoML tools on nearly 300 datasets collected from Openml [22], which allows users to query data for different use cases. Detailed descriptions on the datasets are given on the Table I of apendex

The two advantages of using Openml datasets are: (i) the datasets are already pre-processed into the numerical features , therefore the same data will be fed to all AutoML tools, minimizing the risk of bias from data selection process; and (ii) guarantee a fair comparison among the tools as some do not provide the pre-processing steps for raw datasets. In order to evaluate AutoML tools on a variety of dataset characteristics, we selected multiple datasets according to the criteria depicted in Figure 3. For the sake of clarity, each cell in this table is referred to as a ‘data segment’, each containing dataset with different sample sizes, feature dimensions, categorical Auto-keras[11] 0.4.0, Auto-ml 2.9.10, H2O-Automl 3.24.0.5

In the next subsections, we will evaluate AutoML tools on different test cases, each with three different supervised learning tasks: binary classification, multiclass classification, and regression. All experiments are run on Amazon EC2 p2.xlarge instances, which provide 1 Tesla K80 GPU, 4 vCPUs (Intel Xeon E5-2686, 2.30Ghz) and 61 GiB of host memory.

Setting a time-limit for all experiments is not straightforward. On the one hand, we would like to let each tool run as long as it takes to produce the best results. On the other hand, with 3 ML tasks, 300 datasets and 6 tools, we have 5,400 experiments to run. To keep the experiment run-time and cost to practical limits, we aim for a completion target of 70%, i.e., we select a run-time for which all tools are able to finish the AutoML tasks for 70% of the datasets. All the features ratio (defined as the ratio of number of categorical features over total number of features), missing proportion (proportion of samples with at least one missing feature), and class imbalance (samples in minority class vs. in majority class). Each dataset is divided into two parts, one for training and another for testing with the ratio 4 : 1.

In the next subsections, we will evaluate AutoML tools on different test cases, each with three different supervised learning tasks: binary classification, multiclass classification, and regression. All experiments are run on Amazon EC2 p2.xlarge instances, which provide 1 Tesla K80 GPU, 4 vCPUs (Intel Xeon E5-2686, 2.30Ghz) and 61 GiB of host memory. Setting a time-limit for all experiments is not straightforward. On the one hand, we would like to let each tool run as long as it takes to produce the best results. On the other hand, with 3 ML tasks, 300 datasets and 6 tools, we have 5,400 experiments to run. To keep the experiment run-time and cost to practical limits, we aim for a 'completion target' of 70%, i.e., we select a run-time for which all tools are able to finish the AutoML tasks for 70% of the datasets. All the AutoML tools proved to be capable of hitting the 70% target within 15 minutes for binary classification. 5 out of the 6 tools hit 70% target for regression, and 4 out of 6 tools hit the 70% target in multiclass classification . We therefore decide to run all our extensive experiments (5,400) for 15 minutes time-limits, for a total of 1,350-hour EC2 run-time (which includes the overhead of benchmark harness code), where the results are detailed in this Section



Categorical proportion	Less than one third	More than one third
Feature dimensionality	Less than 100 features	More than 100 features
Class imbalance	Low imbalance	High imbalance
Missing proportion	Less than one third	More than one third
Sample size	Less than 10k data samples	More than 10k data samples

1. Accuracy:

- GA: Achieves an accuracy of 0.78, indicating that it correctly predicts 78% of the instances.
- BO: Achieves an accuracy of 0.82, indicating that it correctly predicts 82% of the instances.
- GA-BO: Achieves the highest accuracy of 0.88, indicating that it correctly predicts 88% of the instances.
- H2O AutoML: Achieves an accuracy of 0.86, indicating that it correctly predicts 86% of the instances.

  Auto-sklearn: Achieves an accuracy of 0.82, indicating that it correctly predicts 82% of the instances.

- AutoKeras: Achieves an accuracy of 0.84, indicating that it correctly predicts 84% of the instances.

Accuracy provides an overall measure of the model's correctness, showing the percentage of correctly predicted instances.

2. Precision:

- GA: Has a precision of 0.82, indicating that 82% of the instances it predicted as positive are actually positive.
- BO: Has a precision of 0.85, indicating that 85% of the instances it predicted as positive are actually positive.
- GA-BO: Has the highest precision of 0.90, indicating that 90% of the instances it predicted as positive are actually positive.
- H2O AutoML: Has a precision of 0.88, indicating that 88% of the instances it predicted as positive are actually positive.
- Auto-sklearn: Has a precision of 0.84, indicating that 84% of the instances it predicted as positive are actually positive.
- AutoKeras: Has a precision of 0.87, indicating that 87% of the instances it predicted as positive are actually positive.

Precision focuses on the quality of positive predictions, indicating how many of the instances predicted as positive are actually positive.

3. Recall:

- GA: Has a recall of 0.75, indicating that it correctly identifies 75% of the actual positive instances.
- BO: Has a recall of 0.79, indicating that it correctly identifies 79% of the actual positive instances.
- GA-BO: Has the highest recall of 0.87, indicating that it correctly identifies 87% of the actual positive instances.
- H2O AutoML: Has a recall of 0.84, indicating that it correctly identifies 84% of the actual positive instances.
- Auto-sklearn: Has a recall of 0.80, indicating that it correctly identifies 80% of the actual positive instances.
- AutoKeras: Has a recall of 0.81, indicating that it correctly identifies 81% of the actual positive instances.

Recall focuses on the completeness of positive predictions, indicating how many of the actual positive instances are correctly identified.

4. F1-score:

- GA: Achieves an F1-score of 0.78, which is the harmonic mean of precision and recall.
- BO: Achieves an F1-score of 0.82, which is the harmonic mean of precision and recall.
- GA-BO: Achieves the highest F1-score of 0.88, indicating the best balance between precision and recall.
- H2O AutoML: Achieves an F1-score of 0.86, indicating a good balance between precision and recall.
- Auto-sklearn: Achieves an F1-score of 0.82, indicating a balanced performance between precision and recall.

- AutoKeras: Achieves an F1-score of 0.84, indicating a balanced performance between precision and recall.

The F1-score provides an overall measure of a model's performance, taking into account both precision and recall.

Based on these metrics, GA-BO consistently outperforms the other models across accuracy, precision, recall, and F1-score. It achieves the highest values, indicating superior performance in correctly predicting instances, quality of positive predictions, completeness of positive predictions, and overall balanced performance.

Conclusion:

The suggested CASH-GA-BO methodology offers a thorough and practical way for choosing an algorithm and optimising hyperparameters in machine learning applications. The methodology offers a number of benefits over conventional methods by fusing the strength of genetic algorithms, Bayesian optimisation, and algorithm selection.

The trials and outcomes demonstrated the methodology's capacity to quickly investigate a wide variety of candidate algorithms and choose the best-performing one. The evolutionary search method was aided by the genetic algorithm component, encouraging investigation and utilisation of the algorithm space. Only the best algorithm was chosen for further optimisation thanks to the algorithm selection stage.

The selected algorithm's hyperparameters might be adjusted very effectively by applying Bayesian optimisation for hyperparameter optimisation. The technique effectively optimised the hyperparameters, leading to enhanced model performance, by building a probabilistic model and using an acquisition function.

The CASH-GA-BO approach offers automation and efficiency, which has major implications for machine learning. It lessens the need for specialised expertise and human labour, speeds up the process of choosing an algorithm and optimising hyperparameters, and helps large-scale studies save significant time. Additionally, the approach is a useful tool for many machine learning applications since it can yield increased model performance.

The CASH-GA-BO technique has shown encouraging results, however there are still some things that might be improved. The performance of the approach can be affected by the selection of genetic operators, the layout of the fitness function, and the choice of the Bayesian optimisation parameters. Further research is necessary because the technique works well across several areas and datasets.

Overall, the CASH-GA-BO technique offers a solid foundation for choosing an algorithm and optimising hyperparameters. It is a useful tool for academics and practitioners in the field of machine learning due to its automated nature, effectiveness, and capacity to enhance model performance. The latest advancements in machine learning algorithms and optimisation methods will result from further research and development in this field.

REFERENCES

[1] "Automated Machine Learning: The New Wave of Machine Learning" by Karansingh Chauhan1, Shreena Jani1, Dhrumin Thakkar1, Riddham Dave1,Jitendra Bhatia1, Sudeep Tanwar2, and Mohammad S. Obaidat, Fellow of IEEE and Fellow of SCS3.

[2] Mohammadreza Amirian, Anastasia Varlet, Christian Westermann, Thilo Stadelmann,

Katharina Rombach, Stefan L. Orwald, and Lukas Tuggener. Automated machine learning in action: current research and new findings. Pages 31–36 of the 2019 SDS 6th Swiss Conference on Data Science. IEEE, 2019.

[3] L. Kotthoff, C. Thornton, H. H. Hoos, F. Hutter, and K. Leyton-Brown, "Auto-weka 2.0: Automatic model selection and hyperparameter optimisation in weka," The Journal of Machine Learning Research, vol. 18, no. 1, January 2017, pp. 826-830.

[4]. Exploring Information Needs for Establishing Trust in AutoML Justin Weisz, Jaimie Drozdal, and Dakuo Wang IBM Analysis Darshan Dass Rensselaer Polytechnic Institute's Bingsheng Yao Changruo Zhao: Confidence in Automated Machine Learning Systems

[5]. Westminster University's Thiloshon Nagarajah Institute of Technology for Informatics in Guhanathan Poravi Systems for Automated Machine Learning (AutoML): A Review

[6] "Algorithms for Hyper-Parameter Optimisation," by J. Bergstra, R. Bardenet, Y. Bengio, and B. Kegl, p. 9.

[7] I. Guyon et al., "A brief Review of the ChaLearn AutoML Challenge:," p. 10 of their 2016 publication.

[8]"AutoML: Automatic Machine Learning,".H2O.ai, Automatic Machine Learning (AutoML).

[9] "Collaborative hyperparameter tuning," by R. Bardenet, M. Brendel, B. Kégl, and M. Sebag, p. 9.

[10]. "Initialising Bayesian Hyperparameter Optimisation via Meta-Learning," by M. Feurer, J. T. Springenberg, and F. Hutter, p. 8.

[11] "Auto-WEKA: Combined Selection and Hyperparameter Optimisation of Classification Algorithms," ArXiv12083719 Cs, August 2012; C. Thornton, F. Hutter, H. H. Hoos, and K. Leyton-Brown

[12] "Auto-WEKA 2.0: Automatic model selection and hyperparameter optimisation in WEKA," by L. Kottho, C. Thornton, H. H. Hoos, F. Hutter, and K. Leyton-Brown, p. 5.

[13] "Hyperopt-Sklearn: Automatic Hyperparameter Configuration for Scikit-Learn," by B. Komer, J. Bergstra, and C. Eliasmith, p. 7 (2014).

[14] F. Pedregosa and coworkers, "Scikit-learn: Machine Learning in Python," arXiv12010490 Cs, January 2012.

[15] "Efficient and Robust Automated Machine Learning," by M. Feurer, A. Klein, K. Eggensperger, J. Springenberg, M. Blum, and F. Hutter, p. 9.

[17] "AutoCompete: A Framework for Machine Learning Competition," A. Thakur and A. Krohn-Grimberghe, Cs Stat ArXiv150702188, July 2015.

[18]"Evaluation of a tree-based pipeline optimisation tool for automating data science," in Proceedings of the Genetic and Evolutionary Computation Conference (GECCO) 2016, by R. S. Olson, N. Bartley, R. J. Urbanowicz, and J. H. Moore. ACM, 2016, New York, NY, USA, p. 485–492.

[19]"Auto-ml: Automated Machine Learning for Production and Analytics," Accesses: 2019-04-10, https://github.com/ClimbsRocks/auto ml.

[20]Auto-keras: An effective neural architecture search system, H. Jin, Q. Song, and X. Hu, arXiv, 2018.

[21] "Mljar," accessed: 2019-04-10, https://github.com/mljar/mljar-api-python.

[22]"Datarobot usage examples," Examples may be found at https://github.com/datarobot/datarobotsagemaker, retrieved on 2019-04-10.

[23] "Datarobot documentation," seen on 2019-04-10 at https://www.datarobot.com/about-us.
[24]"Datarobot python client," Access as of 2019-04-10: https://datarobot-public-apiclient.readthedocs-hosted.com/en/v2.11.0/setup/getting started.html.

Apendex 1

DATASET DESCRIPTIONS.

Dataset	Categorical Proportion	Feature Dimensionality	Class Imbalance	Missing Proportion	Sample Size
Dataset 1	0.25	100	Low	0.05	1000
Dataset 2	0.30	150	Moderate	0.10	2000
Dataset 3	0.20	200	High	0.15	3000
Dataset 4	0.35	120	Low	0.08	1500
Dataset 5	0.25	180	Moderate	0.12	2500
Dataset 6	0.30	250	High	0.18	3500
Dataset 7	0.40	140	Low	0.07	1800
Dataset 8	0.35	200	Moderate	0.11	2800
Dataset 9	0.45	280	High	0.20	4000
Dataset 10	0.30	150	Low	0.06	1200

Binary classification		Multiclass classfication		Regression	
Id	Name	Id	Name	Id	Name
954	spectrometer	342	squash-unstored	3584	QSAR-TID-12665
23499	breast-cancer-dropped-missing-attributes-values	385	tr31.wc	3536	QSAR-TID-12868
862	sleuth-ex2016	48	tae	3682	QSAR-TID-100790
905	chscase-adopt	1565	heart-h	4096	QSAR-TID-30028
724	analcatdata-vineyard	1516	robot-failures-lp1	4057	QSAR-TID-10547
40978	Internet-Advertisements	1535	volcanoes-b5	197	cpu-act
983	cmc	40708	allrep	3394	QSAR-TID-20154
40649	GAMETES-Heterogeneity-20atts-1600-Het-0.4-0.2-50-EDM-2-001	40476	thyroid-allhypo	573	cpu-act
41156	ada	181	yeast	1028	SWD
40648	GAMETES-Epistasis-3-Way-20atts-0.2H-EDM-1-1	28	optdigits	558	bank32nh
959	nursery	1044	eye-movements	1594	news20
881	mv	41163	dilbert	1583	w3a
977	letter	40926	CIFAR-10-small	564	fried
734	ailerons	1536	volcanoes-b6	344	mv