AI

MDPI

Article

Evaluating the Performance of Automated Machine Learning (AutoML) Tools for Heart Disease Diagnosis and Prediction

Lauren M. Paladino [1], Alexander Hughes [1], Alexander Perera [1], Oguzhan Topsakal [1] and Tahir Cetin Akinci [2,3,*]

[1] Department of Computer Science, Florida Polytechnic University, Lakeland, FL 33805, USA; lpaladino7605@floridapoly.edu (L.M.P.); ahughes3300@floridapoly.edu (A.H.); aperera3727@floridapoly.edu (A.P.); otopsakal@floridapoly.edu (O.T.)

[2] Winston Chung Global Energy Center (WCGEC), University of California at Riverside (UCR), Riverside, CA 92521, USA

[3] Electrical Engineering Department, Istanbul Technical University (ITU), Istanbul 34469, Turkey

* Correspondence: tahircetin.akinci@ucr.edu

Abstract: Globally, over 17 million people annually die from cardiovascular diseases, with heart disease being the leading cause of mortality in the United States. The ever-increasing volume of data related to heart disease opens up possibilities for employing machine learning (ML) techniques in diagnosing and predicting heart conditions. While applying ML demands a certain level of computer science expertise—often a barrier for healthcare professionals—automated machine learning (AutoML) tools significantly lower this barrier. They enable users to construct the most effective ML models without in-depth technical knowledge. Despite their potential, there has been a lack of research comparing the performance of different AutoML tools on heart disease data. Addressing this gap, our study evaluates three AutoML tools—PyCaret, AutoGluon, and AutoKeras—against three datasets (Cleveland, Hungarian, and a combined dataset). To evaluate the efficacy of AutoML against conventional machine learning methodologies, we crafted ten machine learning models using the standard practices of exploratory data analysis (EDA), data cleansing, feature engineering, and others, utilizing the sklearn library. Our toolkit included an array of models—logistic regression, support vector machines, decision trees, random forest, and various ensemble models. Employing 5-fold cross-validation, these traditionally developed models demonstrated accuracy rates spanning from 55% to 60%. This performance is markedly inferior to that of AutoML tools, indicating the latter's superior capability in generating predictive models. Among AutoML tools, AutoGluon emerged as the superior tool, consistently achieving accuracy rates between 78% and 86% across the datasets. PyCaret's performance varied, with accuracy rates from 65% to 83%, indicating a dependency on the nature of the dataset. AutoKeras showed the most fluctuation in performance, with accuracies ranging from 54% to 83%. Our findings suggest that AutoML tools can simplify the generation of robust ML models that potentially surpass those crafted through traditional ML methodologies. However, we must also consider the limitations of AutoML tools and explore strategies to overcome them. The successful deployment of high-performance ML models designed via AutoML could revolutionize the treatment and prevention of heart disease globally, significantly impacting patient care.

Keywords: AutoML; machine learning; cardiovascular disease; coronary artery disease; diagnosis; heart disease; prediction; AutoGluon; AutoKeras; PyCaret

check for updates

Citation: Paladino, L.M.; Hughes, A.; Perera, A.; Topsakal, O.; Akinci, T.C. Evaluating the Performance of Automated Machine Learning (AutoML) Tools for Heart Disease Diagnosis and Prediction. *AI* **2023**, *4*, 1036–1058. https://doi.org/10.3390/ai4040053

Academic Editor: Tim Hulsen

Received: 13 September 2023
Revised: 24 November 2023
Accepted: 27 November 2023
Published: 1 December 2023

1. Introduction

The term "cardiovascular disease" (CVD) applies to any disorder affecting the cardiovascular system (heart and blood vessels) [1]. Over 17 million people die from CVD annually globally [2], and heart disease specifically is the leading cause of death in the United States, killing almost 700,000 people in 2020 [3]. Atherosclerosis, or the buildup of plaque within the arteries, leads to coronary artery disease (CAD), one of the most common types of heart disease [3]. Risk factors for CVD include obesity, hypertension, hyperglycemia, and "high alcohol intake" [4]. Doctors usually diagnose CAD through a combination of physical examination, family history, and diagnostic tests including angiography, a type of contrast X-ray that measures the extent of narrowing of the blood vessels [5–7]. Other diagnostic tools include electrocardiography, sonography, and blood testing [4]. The public health importance of addressing heart disease, along with the abundant and continuously growing data, means that machine learning (ML) techniques could be utilized to find meaningful patterns in clinical data and predict the presence of heart disease. Numerous researchers have explored applying ML to medical diagnosis using many different techniques and ensembles, as discussed in the Related Work section below. However, to apply ML, a certain level of computer science knowledge is required, which may be a barrier to widespread use by healthcare professionals [8]. Steps in a typical machine learning project include framing the problem, obtaining the data, conducting exploratory data analysis, preparing the data, exploring the different models, and fine-tuning the models. AutoML tools allow for the implementation of complex models, including feature engineering and hyperparameter optimization, requiring fewer lines of code and less technical knowledge than traditional ML methods. AutoML can automate the steps of data preparation, model selection, and the fine-tuning of models. By utilizing AutoML frameworks, it is possible to develop a cost-effective way to predict heart disease, providing health professionals with a powerful tool for heart disease prediction and diagnosis [4]. AutoML frameworks are machine learning tools that automate many of the more complex machine learning processes to allow non-experts access. This enables someone with less technical knowledge and with fewer lines to use powerful machine learning algorithms. AutoML tools provide access to a range of ML models that make implementing machine learning applications using various datasets much easier. Most AutoML tools are capable of data preprocessing, hyperparameter tuning, and model training automatically without extensive code or data manipulation [9–11].

This study aims to assess the efficacy of automated machine learning (AutoML) tools in the diagnosis of heart disease—a domain where, to our current knowledge, there is yet to be a comprehensive comparative analysis of various AutoML frameworks. We have conducted a thorough investigation of three widely used AutoML Python libraries—PyCaret, AutoGluon, and AutoKeras—across three distinct datasets: Cleveland, Hungarian, and a synthesized dataset amalgamating four separate databases.

Given the grave implications of cardiovascular diseases, and with heart disease at the forefront as a major global health concern, the need for advanced, accurate diagnostic methods is more pressing than ever. Our research delves into the potential of AutoML tools to meet this need, scrutinizing their capabilities in contrast to traditional machine learning techniques. For a robust comparison, we meticulously engineered ten machine learning models using conventional processes, such as exploratory data analysis (EDA), data cleansing, and feature engineering, applying the comprehensive tools provided by the sklearn library.

The ambition of our work is to illuminate the strengths and possible applications of AutoML in refining the diagnosis of heart disease, thereby contributing meaningfully to both the healthcare sector and the field of machine learning. Through this exploration, we intend to delineate the extent to which AutoML tools can not only streamline the diagnostic process but also potentially increase its precision, offering a significant step forward in combating this global health challenge.

In the following sections, we provide a synthesis of pertinent literature, laying the groundwork for the context of our research. Subsequently, we elucidate the methodology adopted in our study, detailing the selection and application of AutoML tools, as well as the datasets chosen for evaluation. Within our methodological exposition, we articulate the conventional techniques implemented to construct a high-performing model manually.

We then proceed to delineate the results obtained, paving the way for an in-depth discussion that interprets the findings within the broader scope of current knowledge and practice. Finally, we encapsulate the study by highlighting its key contributions and the significance of our findings, drawing attention to the implications of our work in the field of heart disease diagnosis through advanced machine learning technologies.

2. Related Works

Medical data analysis and prediction is a critical research area, and dozens of research groups have already applied non-automated ML techniques to heart disease specifically. Hazra et al., Khan et al., and Marimuthi et al. have reviewed and summarized some of the extant research and the accuracies achieved [12–14]. Applied techniques include artificial neural networks (ANN), decision trees, K-nearest neighbor, naïve Bayes, logistic regression, support vector machines (SVM), and association rules, with most researchers obtaining high accuracies [12–14]. Decision trees, ANN, and SVM are three of the most frequently used methods, and many groups had improved success using ensembles of multiple methods [12–14]. Nagavelli et al. compared several different machine learning algorithms in 2022 in their paper titled "Machine Learning Technology-Based Heart Disease Detection Models." The authors of the paper used SVM, the naïve Bayes weight approach, and XGBoost algorithms. The Cleveland dataset and the Statlog dataset, which is a smaller version of the Cleveland, were used for the machine learning algorithms. DBSCAN was used to remove outlier data, and Python library XGBoost V0.81 was used to implement the XGBoost portion of the algorithm. Their results showed that XGBoost had the highest accuracy, with lower accuracy for the SVM and naïve Bayes approaches [15].

In another paper, titled "Heart Disease Diagnosis and Prediction Using Machine Learning and Data Mining Techniques: A Review", the authors reviewed a list of relevant machine learning information [10]. The paper explained several different machine learning algorithms that have been used in heart disease datasets, including decision trees, K-means, SVM, naïve Bayes, artificial neural networks (ANN), Iterative Dichotomiser 3 (ID3), classification and regression trees (CART), random forest, a-priori, fuzzy logic, and association rules. During this review, several tools and environments for data learning were examined, including WEKA (Waikato Environment for Knowledge Learning), RapidMiner, TANAGRA, Apache Mahout, MATLAB, Java, C, and Orange. These machine learning algorithms and tools were extracted from a survey of 35 research papers and represent what has been used in non-automated machine-learning research for heart disease [12]. Various software systems (e.g., WEKA, RapidMiner, TANAGRA) and programming languages (e.g., Java, MATLAB, Python) are available for the implementation of ML models. WEKA, based on Java, was one of the more commonly referenced tools in the papers by Hazra et al., Khan et al., and Marimuthi et al. [12–14]. Singh et al. used WEKA for predicting heart disease, with a dataset of 303 records and a multilayer perceptron neural network (MLPNN) with backpropagation [6,16–19].

It is important to note that new technologies are emerging for disease diagnosis including heart disease diagnosis. For example, hyperspectral and multispectral imaging systems are non-invasive diagnostic tools that capture and analyze a wide spectrum of light to identify, assess, and map various biological materials. These systems are increasingly being applied in the field of medical diagnosis, including for the detection and analysis of diseases [16,17]. The data from these systems are also being utilized in machine/deep learning models for improved diagnosis [18,19].

Pol et al. used Python and the AutoML tool PyCaret to predict the presence or absence of heart disease [8]. Padmanabhan et al. used Python and Auto-Sklearn on the same Cleveland Heart Disease dataset as Pol et al. [20]. Valarmathi and Sheela used the AutoML tool TPOT with the Cleveland dataset, but only when tuning the hyperparameters of their random forest and XG boost classifier models [21]. With the growth of AutoML tools, there has been a concern regarding if AutoML is comparable to previous machine learning methods. A paper titled "Physician-Friendly Machine Learning: A Case Study with Cardiovascular Disease Risk Prediction" compared the performance of the AutoML tool Auto-Sklearn to that of more traditional manual machine learning techniques for accuracy [20]. In the paper, a graduate student who was experienced in creating machine learning models was allowed one month to develop models using manual techniques using the scikit-learn python library. Then, models were created using the Auto-Sklearn tool to automatically create the model, which took around 30 min to complete and only required four lines of Python code. The results showed comparable results of the models. The paper concluded that their research strongly suggests that AutoML is a useful approach that allows less experienced users to quickly create models that are competitive and comparable to models created by experienced machine learning users [20]. Pol et al. used another popular Python AutoML tool, PyCaret, in their paper titled "AutoML: Building A Classification Model with PyCaret." PyCaret was used with a heart disease dataset and trained on a 70/30 train/test split with normalization turned on. PyCaret was then used to train the dataset with all algorithms available in its library. The paper concluded the results were "favorable" for classification with logistic regression for their heart disease dataset [8].

Romero et al. benchmarked the performance of the AutoML tools AutoSklearn, H2O, and TPOT in disease prediction. They used medical claims data of more than 12 million people to predict six different diseases. These did not include heart disease. While the dataset used was large, the disease prevalence was very low, with the highest prevalence being for chronic kidney disease at 0.63%. They found that the performance of the different tools varied for different diseases, with prostate cancer prediction having some of the highest accuracies and type 2 diabetes some of the lowest. Performance between the tools themselves also varied. H2O produced some of the most accurate models across the diseases examined, though Romero et al. noted that the variation between the tools was not large [22].

In their 2021 paper, Ferreira et al. [9] compared the performance of eight open-source AutoML tools. Like us, they used default settings for the different tools when possible. Though they did not use the same datasets as we did, the tools they used included AutoGluon and AutoKeras, two of the three used in this paper. AutoGluon was significantly faster than the other tools. AutoKeras was one of the fastest deep learning/neural network tools that they used. While it was the slowest tool used here, it was the only deep learning tool used in this study. Ferreira et al. found that no single AutoML tool performed better than the others across all datasets [9].

3. Methodology

In the following sections, available AutoML tools are introduced first, followed by the ones included in this study and the inclusion criteria. Then, the existing datasets and the details of the selected datasets are given, and the selected performance criteria are also introduced in detail.

3.1. AutoML Tools

There are several open-source and proprietary AutoML tools available. The following well-known AutoML tools (listed in alphabetical order) were considered for our study. AutoGL is open source and was created at Tsinghua University for AutoML on graphs and contains four modules, including auto feature engineering, model training, hyperparameter optimization, and auto ensemble [23,24]. AutoGluon is an open-source tool created by Amazon that can automate machine learning and deep learning algorithms for text, images, and datasets [25–27]. AutoGluon evaluates and compares a variety of models and assists in selecting the best model to utilize and fine-tune. It is developed to support very specific problem types: regression and classification using tabular data, image classification, and object detection. AutoGluon provides a user-friendly interface and tools that allow for data to be trained effectively within a single line of code as it automatically balances efficiency and performance, allowing for less headaches with hyperparameter editing [28]. AutoKeras is an AutoML system based on Keras and TensorFlow and was developed at DATA Lab at the Texas AM campus [29–31]. AutoKeras allows for the building and training of deep neural networks and automates the process of hyperparameter tuning and model selection with an easy-to-use interface [29]. Auto-Sklearn was built around scikit-learn and automatically searches for the best machine learning algorithm for a dataset, along with hyperparameter optimization [32,33]. Auto-Sklearn provides efficient processes to learn the data and continue learning from similarly identified datasets through "meta-learning" and a Bayesian optimizer, which learns from the preprocessed data, features, and classifier to determine the best model approach [34]. While Auto-Sklearn can be an effective approach, reliance on the data being large enough is a necessity [34,35]. MLBox is a library for Python that offers powerful AutoML tools and predictive models for classification and regression. It can use deep learning, stacking, and LightGBM and offers interpretations of prediction models [36]. Neural Network Intelligence is an automated machine learning toolkit created by Microsoft that searches for the best hyperparameters and neural architecture by running trial jobs automatically [37]. PyCaret provides an ideal experience for productivity and low-effort ML solutions and designs and launches quick prototypes [38,39]. PyCaret quickly tests a variety of models, providing the data scientist with an effective understanding of what models work efficiently and accurately between classification and regression tasks [8]. TPOT (tree-based pipeline optimization tool) is another open-source automated machine learning tool for Python, built on top of scikit-learn, that optimizes machine learning pipelines [40]. TPOT is still in active development and can automate many steps of the machine learning process, such as feature selection, feature preprocessing, feature construction, model selection, and parameter optimization [40]. TPOT explores thousands of possible pipelines using genetic algorithms and returns the best one for a given dataset [40,41].

3.2. Details of the Selected AutoML Tools

Effective AutoML modeling choices for the various heart disease datasets were narrowed down to AutoGluon, AutoKeras, and PyCaret. AutoGluon was chosen due to its high performance and ease of use [38]. It evaluates several different machine learning algorithms to find the best model for the data. These algorithms include ExtraTreeEntr, RandomForestEntr, WeightedEnsemble-L2, ExtraTreesGini, RandomForestGini, XGBoost, KneighborsUnif, and KNeighborsDist. Random forest is a machine learning algorithm that uses multiple randomized decision trees (an algorithm based on splitting binary decision nodes) to make predictions. The extra trees algorithm is similar to random forest except that the split in the decision trees is randomly selected. The models with the Gini and Entr suffixes indicate the measures used to determine how a decision tree node splits. These measures are referred to as the Gini index and Entropy, which is a measure of the purity of the split [42]. XGBoost is an extreme gradient-boosted tree with an ensemble algorithm, with each tree boosting misclassified attributes of the previous tree [43]. K-nearest neighbor is a machine learning algorithm that uses classification based on data points that are close to each other. Weighted ensemble algorithms combine multiple model predictions, where each model's contribution is weighted based on how accurate the model is, creating a single model based on the combination [44].

AutoKeras trains deep neural networks and performs model selection and hyperparameter tuning, with little user input required [29]. Some of the highest accuracies achieved by previous research groups predicting heart disease using traditional methods were obtained using neural networks [12,13]. Due to AutoKeras's ease of use and other researchers' success using neural networks, AutoKeras was another tool that was selected. AutoKeras offers classification and regression tools for image, text, and structured data. It also offers a TimeSeriesForecaster and more advanced tools for multi-modal and multi-task analyses and customized model development [29]. Default settings were used for the AutoKeras StructuredDataClassifier, including max-trials = 100, epochs = 1000, and validation-split = 0.2.

PyCaret, being built from the sklearn groundwork, allows for more adaptability and model evaluation but relies more on the strength of the dataset and preprocessing preparation. PyCaret operates by applying multiple machine learning models and algorithms to either preprocessed or unprocessed data to determine which machine learning model best applies to the data given and grants the most accurate model to utilize [39]. A variety of models are used within PyCaret [45], which include logistic regression, quadratic discriminate analysis (QDA), light gradient boosting machine, linear discriminant analysis (LDA), SVM, naïve Bayes (NB), and several other classifiers. PyCaret functions with a user setup that establishes what the target variable for prediction is amongst all models that are given within the PyCaret utility. PyCaret has various performance metrics embedded that allowed the authors to compare the various algorithms, including a confusion matrix, class prediction error, and precision–recall curve [8]. Post-data preprocessing enables significantly better results within the model evaluation step and leads to model tuning, which further refines the best-selected model and prepares it for prediction analysis [46]. The prediction function best operates after the best-selected model undergoes tuning and then proceeds to test the given model with the test data that split after data preprocessing. PyCaret does not just focus on singularly labeled data but also works within multiclass data, which expands the capacity that it can operate with and provides further data analysis [47].

The selection criteria prioritized open-source tools that were compatible with the latest version of Python. Compatibility ensured that the selected tools were actively maintained and could be seamlessly used in Google Colab, our chosen analysis environment. To provide a diverse range of approaches, we aimed to include a variety of tools. Auto Gluon is predominantly based on decision tree methods, while Pycaret incorporates other machine learning algorithms such as QDA, LDA, SVM, and NB, along with ensembles. Given the limited availability of deep learning approaches for small datasets, only one neural network tool, AutoKeras, was included [9,35]. Although it is possible to use the tools collectively, in this study, we evaluated each tool as an independent solution.

3.3. Dataset

Multiple resources are available for datasets, such as Google [48], IEEE [49], Mendeley [48–50], Kaggle [50], and the University of California, Irvine (UCI) [51]. However, patient privacy is an important consideration when handling health data, with HIPAA requiring “IRB waiver or patient authorization for research” use of protected health information [52–55]. Some datasets are freely accessible, while others require a research request or an access fee. Two open-access heart disease datasets available from UCI are the Cleveland Heart Disease [54] and the Statlog (Heart) datasets [56]. Most of the previous studies utilizing ML to predict heart disease used the Cleveland Heart Disease dataset when training and testing their models, including Marimuthu et al. [14], Pol et al. [8], Valarmathi et al. [20], Padmanabhan et al. [20], six research groups reviewed by Hazra et al. [9], and three different research groups surveyed by Khan et al. [12]. Researchers Dangare et al. [5], El Bialy et al. [55], Nagavelli et al. [15], Sarra et al. [56], and Ahmed [57] used both the Cleveland and Statlog datasets in their analyses.

3.4. Details of the Selected Datasets

The Cleveland and Statlog datasets were both considered for use in this research due to them being easily accessible and having been used by other researchers, allowing for direct comparisons of results. However, when performing preliminary exploratory analysis, we observed very similar patterns in the histograms and attribute distributions. This suggested the possibility that the Statlog dataset is a subset of the Cleveland dataset, despite other researchers having used both datasets in their analyses. The source of the Statlog dataset is not clear in its documentation, so we compared the datasets and confirmed that Statlog is a subset of Cleveland, making it unsuitable and redundant for our use. The second dataset chosen was the Hungarian Heart Disease dataset, which is available from the same UCI repository location as Cleveland. The Cleveland and Hungarian datasets are of a similar size, with 303 and 294 observations, respectively, but Cleveland is more complete, with fewer missing values: 6 versus 781. Besides these two datasets, two additional datasets, UCI: Switzerland (123 samples) and Long Beach, CA (V.A. Medical Center) (200 samples) were utilized to form a larger third dataset containing 920 total samples. All datasets use the same 13 attributes and a label, as shown in Table 1. General statistics for the datasets are in Tables 2 and 3. Distributions for the different attributes for the combined dataset are shown in Figure 1, 2 and 3 and are separated by heart disease status: HD positive versus HD negative.

Table 1. Attribute descriptions of datasets.

Attribute	Description
Age	Age in years
Sex	Sex (1 = male; 0 = female)
Cp	Chest pain type
Trestbps	Resting blood pressure (in mm Hg on admission to hospital)
Chol	Serum cholesterol in mg/dL
Fbs	Fasting blood sugar > 120 mg/dL (1 = true; 0 = false)
Restecg	Resting electrocardiographic results
Thalach	Maximum heart rate achieved
Exang	Exercise-induced angina (1 = yes; 0 = no)
Oldpeak	ST depression induced by exercise relative to rest
Slope	The slope of the peak exercise ST segment
Ca	Number of major vessels (0–3) colored by fluoroscopy
Thal	3 = normal; 6 = fixed defect; 7 = reversible defect
Num	Diagnosis of heart disease (angiographic disease status)

Table 2. Number of missing values per attribute by dataset.

Attribute	Cleveland	Hungarian	Switzerland	VA
Trestbps	0	1	2	56
Chol	0	23	0	7
Fbs	0	8	75	7
ReThalach	0	1	1	53
Exang	0	1	1	53
Oldpeak	0	0	6	53
Slope	0	190	17	102
Ca	4	290	118	198
Thal	2	266	52	166

Attributes not shown were complete.

Table 3. Attribute correlations with label.

Attribute	Cleveland	Hungarian	Combined
Thalach	−0.417167	−0.331074	−0.385972
Fbs	0.025264	0.162869	dropped
Chol	0.085164	0.202372	-0.234679
Trestbps	0.150825	0.139582	0.103828
Restecg	0.169202	−0.031988	0.062304
Age	0.223120	0.159315	0.282700
Sex	0.276816	0.272781	0.307284
Slope	0.339213	dropped	dropped
Cp	0.414446	0.505864	0.471712
Oldpeak	0.424510	0.545700	0.373382
Exang	0.431894	0.584541	0.443433
Ca	0.460033	dropped	dropped
Thal	0.522057	dropped	dropped

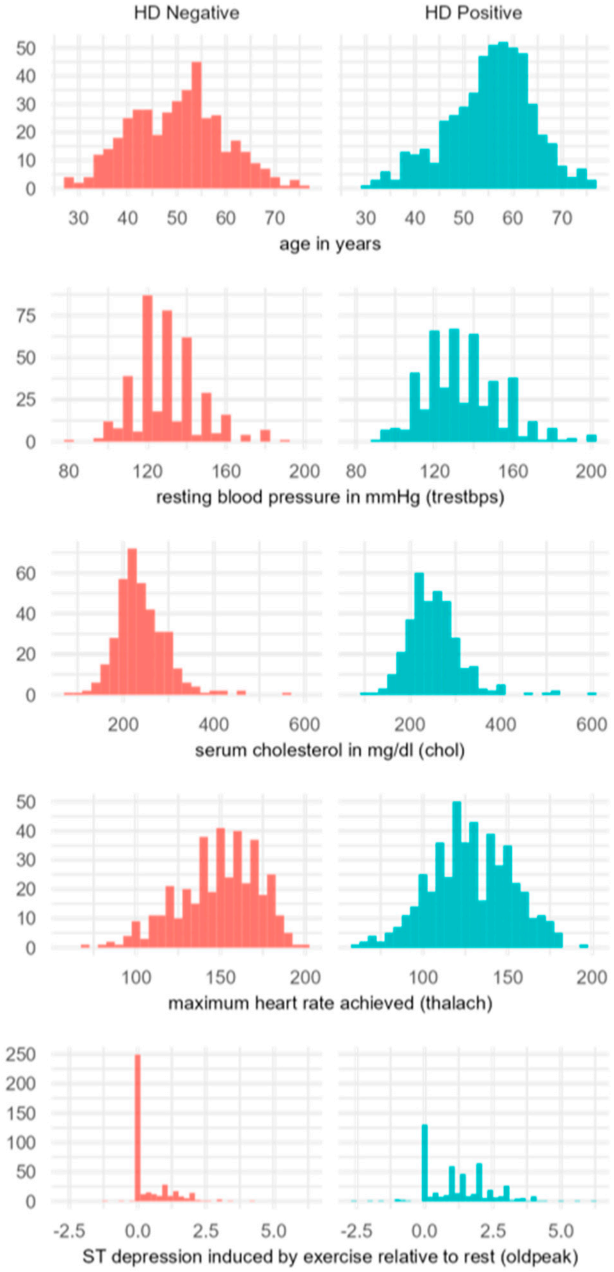

Figure 1. Histograms of five continuous attributes from the combined dataset, filtered for heart disease (+) positive and (−) negative patients.

Table 2 shows the number of missing values per attribute for each of the datasets used. Age and sex are omitted from the table as they were complete in all datasets. The number of missing values varied between datasets, with Cleveland being the most complete. Certain attributes, like ca, thal, and slope, were excluded from analyses using the Hungarian, Switzerland, or VA datasets due to very few observations being available. In addition to the number of major vessels colored by fluoroscopy (ca), thalassemia, and slope, fasting blood sugar (fbs) was also excluded from the combined dataset due to the large number of missing values in the Switzerland dataset.

Histograms of the continuous attributes are shown in Figure 1, with heart disease-positive patients on the right and negative on the left. The histograms clearly show the differences between heart disease-positive and -negative patients. First of all, when the age data were examined, it was seen that the age of positive patients was higher than that of negative patients. This situation reflects that positive patients' age distribution is skewed compared to negative patients. It was observed that trestbps (resting blood pressure) data were generally reported in ten-unit increments. While it is noteworthy that the trestbps values of negative patients were close to values such as 120, 130, and 140, an abnormal outlier value of 0 was observed in positive patients. In Chol (serum cholesterol) data, there were many outliers close to 0 in positive patients, while there were fewer outliers in negative patients. These outliers were not specified in the dataset documentation, with missing data denoted by "−9" or NaN. However, it is physically impossible for cholesterol or blood pressure to be 0. Considering these outliers, the trestbps and cholesterol data appeared to have similar skewness and kurtosis characteristics in positive patients compared to negative patients, with higher mean values in positive patients. While Thalach (maximum heart rate reached) data showed an approximately symmetrical distribution for positive patients, the distribution was negatively skewed in negative patients and had a higher mean value. Oldpeak (ST depression caused by exercise) data contained many 0 values. These 0 values appeared to be actual observations and not missing values, given the magnitude of the neighboring observations, but this was not noted in the data documentation or a quick literature search. Additionally, approximately twice as many 0 values were observed in positive patients than in negative patients. These data are presented in detail in Figure 1 and reveal significant differences in the datasets.

Figures 2 and 3 show the differences in different characteristics between heart disease-positive and negative patients. Female patients (0) constituted a significant proportion (about one-third) of the heart disease-negative group, while representing a smaller proportion (about one-tenth) of heart disease-positive cases. The "cp" feature indicated that patients with heart disease predominantly experience asymptomatic chest pain (4), while typical angina (1) was rarely seen in heart disease-negative cases. Compared with other individual features, the differences in "fbs" (fasting blood glucose) and "restecg" (resting electrocardiography results) between the two groups of patients were less pronounced. "Exang" (exercise-induced angina) was rare in heart disease-negative patients but common in more than half of heart disease-positive cases. By understanding these complex patterns between different characteristics, changes in factors affecting the diagnosis of heart disease could be clearly observed, as shown in Figures 2 and 3.

Discrete attributes are shown in Figures 2 and 3. Women (0) were found to make up less than a quarter of all patients but almost a third of heart disease-negative patients and only about a tenth of heart disease (+) positive patients. In analysis, this makes sex an important feature. Results obtained from analyses where data are not combined with sex may be very different. Cp describes chest pain. Heart disease patients were found to predominantly exhibit asymptomatic chest pain (4), while little typical angina (1) was seen among negative patients. The differences in fasting blood sugar (fbs) and resting electrocardiographic results (restecg) between heart disease (+) positive and (−) negative patients were not as large as those for other discrete attributes. Very few heart disease (−) negative patients were found to experience exercise-induced angina (exang), while more than half of positive patients did.

The slope was flat (2) for most positive patients and rarely downsloping (3) for negative patients. The number of major vessels colored by fluoroscopy, ca, was seen to be predominantly 0 for negative patients, but it is not known if this value is an observation or only represents a missing observation. It was observed most heart disease (−) negative patients were normal for thalassemia, and most heart disease-positive patients had a reversible defect. Num was the label, for which 0 indicated heart disease (−) negative and 1 through 4 indicated heart disease-positive. All prior research we found using the Cleveland dataset used a binary label. We chose to do the same for our analyses.

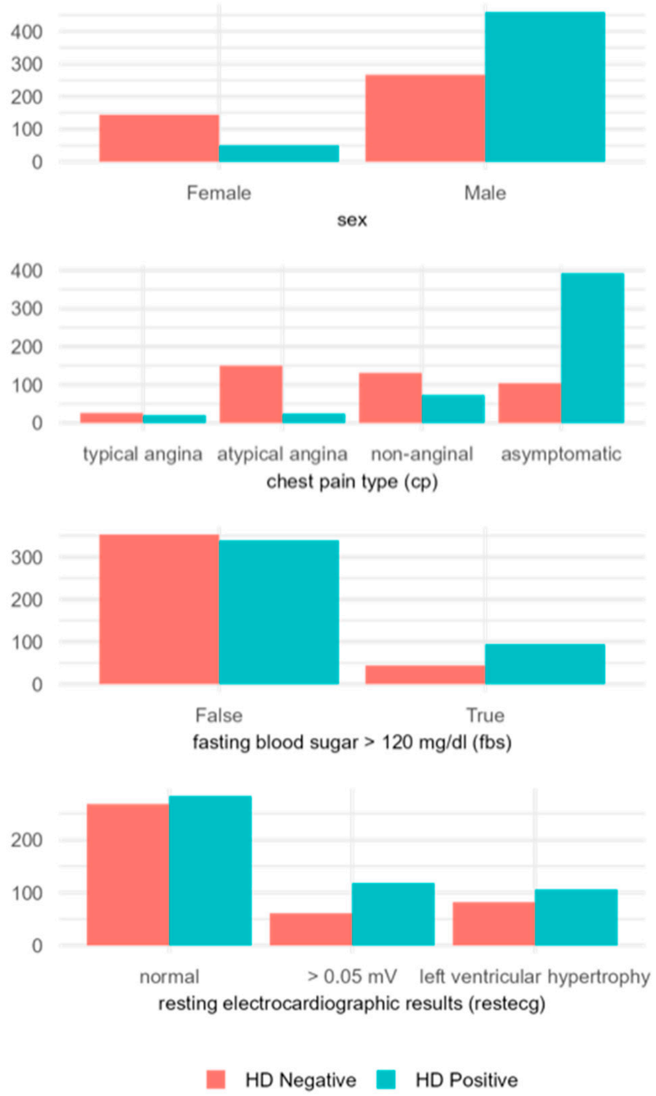

Figure 2. Distributions of the first four of the eight discrete attributes from the combined dataset, filtered for heart disease (+) positive and (−) negative patients, excluding the binary label.

Table 3 shows the correlations of the different attributes with the dependent variable, num. The attributes chosen for analysis for each of the datasets are shown with unused features marked as "dropped". The number of major vessels colored by fluoroscopy (ca) and thalassemia were both highly correlated with num in the Cleveland dataset, meaning their exclusion in the Hungarian and combined datasets may have negatively impacted prediction accuracy. Slope also had a correlation above 0.2 for the Cleveland dataset, though not as high as for the number of major vessels colored by fluoroscopy (ca) or thalassemia. The slope may be worthy of inclusion in future analyses with larger datasets, provided there is an adequate number of observations available, as it has been found to be a useful predictor in other applications [54]. The exclusion of fasting blood sugar (fbs) from the combined dataset was supported by the relatively low correlations in the Cleveland and Hungarian datasets.

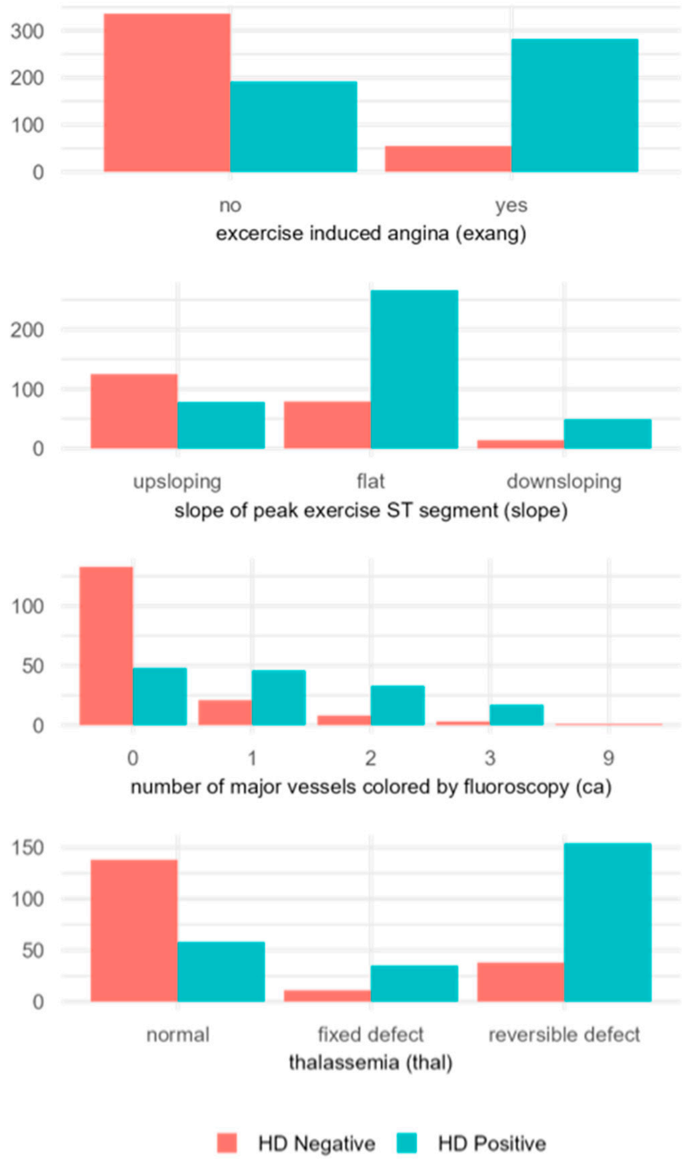

Figure 3. Distributions of the last four of the eight discrete attributes from the combined dataset, filtered for heart disease (+) positive and (−) negative patients, excluding the binary label.

3.5. Performance Metrics Used

Accuracy and F1 score metrics were used to evaluate the performance of the AutoML tools. Accuracy (ACC) is a measure of how well the model predicts the outcome. The formula for accuracy is the number of correct predictions divided by the total number of predictions (Equation (1)). This metric, however, can be biased due to data imbalances and thus lead to skewed results.

$$\text{Accuracy} = \frac{(TP + TN)}{(TP + TN + FP + FN)} \tag{1}$$

One metric that takes this into account and is commonly used for machine learning is the F1 score. The F1 score is the harmonic mean between precision and recall (2 times the product of precision and recall divided by the total sum of precision and recall) (Equation (4)). Precision is defined as the number of true positives divided by the sum of true positives and false positives (Equation (2)). The recall metric is the number of true positives divided by the total number of true positives and false negatives (Equation (3)) [58].

$$\text{Precision} = \frac{TP}{(TP + FP)} \tag{2}$$

$$\text{Recall} = \frac{(TP)}{(TP + FN)} \tag{3}$$

$$\text{F1 score} = 2\frac{(precision \times recall)}{(precision + recall)} \tag{4}$$

3.6. Applying Traditional Steps of Manually Generating the Well-Performing Model

To benchmark the performance of AutoML-generated models against those produced by traditional manual techniques, we meticulously followed a series of steps to develop a classification model using sklearn libraries for heart disease using the Cleveland Heart Disease dataset:

(1). Data Cleaning: Rows with missing information in the "ca" and "thal" columns were removed to ensure data integrity.

(2). Data Type Conversion: All fields were converted to numeric data types to facilitate subsequent analysis and modeling.

(3). Correlation Analysis: The correlations between the fields and the target label were analyzed. Four fields ("chol", "fbs", "trestbps", "restecg") with correlations below 0.2 were identified and subsequently dropped from the dataset.

(4). Data Scaling: The remaining data were scaled to normalize the feature values and ensure comparability across different variables.

(5). Cross-validation: Cross-validation (k = 5) accuracy scores were calculated for 10 different machine learning algorithms. The algorithms used were stochastic gradient descent (SGD), logistic regression, support vector machine with a linear kernel, support vector machine with an RBF kernel, decision tree classifier, random forest classifier, extra trees classifier, AdaBoost classifier, gradient boosting classifier, and XGBoost.

(6). Hyperparameter Tuning: The top-performing algorithms (AdaBoost, rando forest, gradient boosting, XGBoost) were selected for further improvement through hyperparameter tuning. A grid search was performed using various combinations of hyperparameters, including n-estimators (100, 200, 300, 400, 500), learning-rate (0.3, 0.1, 0.05), max-features (1, 0.7, 0.5, 0.4, 0.3), subsample (1, 0.5, 0.3), max-samples (1, 0.5, 0.3, 0.2), and bootstrap (True, False).

(7). Ensemble Voting Classifier: Based on the fine-tuned estimators (AdaBoost, random forest, gradient boosting, XGBoost) and the other top-performing estimators (SVC, SGD, logistic regression), an ensemble voting classifier was constructed. This ensemble classifier combined the predictions of multiple models, leveraging their collective knowledge to make a final classification decision.

By following these steps, we conducted a comprehensive analysis and model selection process to generate the most effective machine learning model utilizing the machine-learning algorithms listed above and forming an ensemble learning model for classifying heart disease using the Cleveland Heart Disease dataset. The outcome of the above steps is shared and compared in the Results section and the code is available on GitHub [58].

4. Results

The datasets were analyzed using PyCaret v3.0, AutoGluon v0.7.0, and AutoKeras v1.1.0. Default settings were used for all AutoML tools. Missing values were imputed to the respective attribute's mode, if discrete, and mean, if continuous. Attributes ca, thal, and slope were excluded from the Hungarian data analyses due to missing more than half of each of these attributes. Data were split with 80% for training and 20% for testing, using stratified sampling by sex. Stratified sampling was used due to the difference in representation by sex. This difference was less pronounced in the Cleveland and Hungarian datasets, but it was more significant in the Switzerland and VA datasets. We verified that the representation of heart disease-positive patients was approximately the same for both the training and testing subsets of the Cleveland, Hungarian, and combined datasets: 46%, 36%, and 55%, respectively. We also verified that the data were successfully split using stratified sampling. The testing and training datasets were both approximately 33%, 27%, and 21% female for the Cleveland, Hungarian, and combined datasets, respectively.

As described in the previous Methodology section, we performed traditional steps to develop a well-performing machine learning model using the sklearn libraries on the Cleveland dataset to compare the manually generated models with the models generated by the AutoML tools. The manual steps included cleaning the data, reducing the data based on correlation analysis, exploring the performance of 10 machine learning algorithms, selecting top-performing models and applying hyperparameter fine-tuning, and then ensembling various top-performing models to form a well-performing model. The results of this manual approach are in Table 4.

Table 4. ML model generation using traditional manual steps on the Cleveland dataset.

Machine Learning Algorithm	Accuracy (Correlated)	Accuracy (Unreduced)
Stochastic Gradient Descent (SGD)	0.59	0.58
Logistic Regression	0.59	0.59
Support Vector Machine (SVM) (Linear Kernel)	0.55	0.57
Support Vector Machine (SVC) (RBF Kernel)	0.57	0.56
Decision Tree	0.52	0.49
Random Forest	0.62	0.59
Extra Trees	0.57	0.57
AdaBoost	0.58	0.57
Gradient Boosting	0.60	0.59
XGBoost	0.55	0.56
Ensemble of the following: AdaBoost, Random Forest, Gradient Boosting, XGBoost, SVM-Linear, SGD, Logistic Regression	0.60	0.58

As shown in Table 4, the best-performing models were based on ensemble algorithms such as random forest, gradient boosting, and our custom ensemble model that combined seven models. When the dataset was processed by reduction based on correlated fields, it tended to perform better, as can be noticed by comparing accuracy scores using the correlated and unreduced datasets in Table 4. Results from analyses using the three different AutoML tools on the three different datasets are in Tables 5 and 6. Data in the "correlated" row are for the outcome performed on datasets using only features that correlated with the label of at least 0.2. The "unreduced" data are for the outcome where all features were used. The top three models from the leaderboard are listed for PyCaret and AutoGluon in Tables 5–7. Results from a single analysis for AutoKeras for each dataset are shown. Accuracies for repeat analyses using AutoKeras are shown in Table 8.

PyCaret accuracies were generally in the low eighties with slightly lower F1 scores. AutoGluon accuracies were the highest of all three AutoML tools. Its accuracies for the unreduced dataset were in the mid-to-high eighties and its F1 scores were the same or higher. The correlated dataset results were slightly lower, in the mid-eighties, except for the results achieved by the top model on the leaderboard, which were only around 78%. The accuracy for the unreduced dataset, when analyzed using AutoKeras, was comparable, about 80%, with a slightly higher F1 score. The AutoKeras accuracy for the Cleveland correlated dataset was only 54%. The F1 score was still poor but substantially better at approximately 67%. Other analyses using the same data and input conditions produced much higher accuracies (see Tables 5–7).

Table 5. Cleveland dataset analysis using PyCaret, AutoGluon, and AutoKeras.

	Cleveland		
	Accuracy	**F1 Score**	**Best Model**
Unreduced:			
	0.8525	0.8037	1. Linear Discriminant Analysis
PyCaret	0.8215	0.7998	2. Ridge Classifier
	0.8180	0.7939	3. Naïve Bayes
	0.8688	0.8709	1. WeightedEnsemble_L2
AutoGluon	0.8688	0.8709	2. RandomForestGini
	0.8524	0.8524	3. RandomForestEntr
AutoKeras	0.8033	0.8182	N/A
Correlated:			
	0.8137	0.8012	1. Logistic Regression
PyCaret	0.8048	0.7814	2. Linear Discriminant Analysis
	0.8008	0.7775	3. Ridge Classifier
	0.7868	0.7796	1. WeightedEnsemble_L2
AutoGluon	0.8524	0.8474	2. RandomForestGini
	0.8360	0.8333	3. RandomForestEntr
AutoKeras	0.5410	0.6667	N/A

Table 6. Combined dataset analysis using PyCaret, AutoGluon, and AutoKeras.

	Combined		
	Accuracy	**F1 Score**	**Best Model**
Unreduced:			
	0.6873	0.6678	1. Logistic Regression
PyCaret	0.6833	0.6839	2. Linear Discriminant Analysis
	0.6832	0.6484	3. Ridge Classifier
	0.8478	0.8691	1. WeightedEnsemble_L2
AutoGluon	0.8478	0.8691	2. RandomForestEntr
	0.8423	0.8651	3. ExtraTreesGini
AutoKeras	0.8152	0.8365	N/A
Correlated:			
	0.7826	0.7311	1. Random Forest Classifier
PyCaret	0.7459	0.7168	2. Ridge Classifier
	0.7432	0.7260	3. Logistic Regression
	0.8423	0.8638	1. WeightedEnsemble_L2
AutoGluon	0.8423	0.8638	2. RandomForestEntr
	0.8369	0.8369	3. RandomForestGini
AutoKeras	0.8315	0.8545	N/A

Table 7. Hungarian dataset analysis using PyCaret, AutoGluon, and AutoKeras.

	Hungarian		
	Accuracy	**F1 Score**	**Best Model**
Unreduced:			
	0.6976	0.6465	1. Logistic Regression
PyCaret	0.6806	0.6092	2. Ridge Classifier
	0.6766	0.6334	3. Linear Discriminant Analysis
	0.8475	0.7804	1.WeightedEnsemble_L2
AutoGluon	0.8474	0.7804	2. RandomForestEntr
	0.8305	0.7619	3. ExtraTreesEntr
AutoKeras	0.8305	0.7059	N/A

Table 7. *Cont.*

	Hungarian		
	Accuracy	**F1 Score**	**Best Model**
Correlated:			
	0.8304	0.7516	1. Ridge Classifier
PyCaret	0.8303	0.7506	2. Log. Regression
	0.8263	0.7470	3. Linear Discriminant Analysis
	0.8983	0.8500	1. WeightedEnsemble_L2
AutoGluon	0.8983	0.8500	2. RandomForestEntr
	0.8644	0.8095	3. ExtraTreesGini
AutoKeras	0.8305	0.7059	N/A

Table 8. Run times and accuracies from repeat analyses using AutoKeras.

		Run 1		**Run 2**		**Run 3**		**Mean σ st. dev.**
		Accuracy	**Run Time**	**Accuracy**	**Run Time**	**Accuracy**	**Run Time**	
Cleveland	Unreduced	0.8197	40 m 51 s	0.7705	6 m 5 s	0.8033	6 m 35 s	0.7978 σ 0.0251
	Correlated	0.7541	18 m 47 s	0.8197	6 m 29 s	0.7705	11 m 10 s	0.7814 σ 0.0341
Hungarian	Unreduced	0.8644	7 m 36 s	0.7966	10 m 29 s	0.6610	52 m 27 s	0.7740 σ 0.1035
	Correlated	0.8136	5 m 34 s	0.8644	7 m 55 s	0.6610	7 m 23 s	0.7797 σ 0.1059
Combined	Unreduced	0.8478	18 m 24 s	0.8207	15 m 57 s	0.8478	12 m 54 s	0.8388 σ 0.0156
	Correlated	0.7989	35 m 51 s	0.8152	10 m 41 s	0.7880	24 m 02 s	0.8007 σ 0.0137

PyCaret's performance on the unreduced Hungarian dataset was significantly worse than that on the Cleveland dataset, with accuracies in the upper 60% and lower F1 scores. Although PyCaret accuracies for the correlated dataset were back in the eighties, the F1 scores were much lower at only about 75%. Accuracies resulting from AutoGluon were all in the mid-to-upper eighties, but its F1 scores were also lower than was seen when using the Cleveland dataset. AutoKeras results were similar, with 83% accuracies and 71% F1 scores.

When analyzing the combined dataset, PyCaret's results were similar to those it achieved when analyzing the Hungarian dataset and lower than those for the Cleveland. Results were below seventy for the unreduced dataset and in the mid-to-low seventies for the correlated.

AutoGluon performed well, with accuracies around 84% and most F1 scores around 86%. AutoKeras results for these runs were also greater than 80%. The code used to obtain the above results is shared on the GitHub page [58]. The open-source code includes exploratory data analysis; AutoML implementation for AutoGluon, PyCaret, and AutoKeras; as well as the code that performed model generation by following the traditional steps.

5. Discussion

The AutoML tools in question function by autonomously testing various embedded algorithms and subsequently presenting the most effective model. As users, we lacked the facility to specify a particular model for application; instead, the tools independently determined the optimal choice. Our comparative analysis, therefore, did not stem from a selection of models we wished to evaluate side by side. Rather, it arose from assessing the peak-performing model provided by one AutoML tool against the counterpart from another, essentially comparing the pinnacle of what each tool asserted to be its most proficient solution.

Our analysis operated under the assumption that tool developers had carefully chosen the best hyperparameters to enhance their tool's performance. Respecting their expertise, we did not modify these parameters, trusting that the default settings were optimized for peak performance. We analyzed each tool in its default state, mirroring typical user experience without advanced machine learning knowledge. This approach provided valuable insights into the performance of each tool with minimal customization—important for users unfamiliar with algorithm intricacies or specific tuning options. AutoML tools have diverse adjustable settings, complicating any attempt to standardize them with identical algorithms. For example, AutoKeras focuses on neural architecture search (NAS) to find the most effective neural network configuration, but it does not inherently support traditional models like SVM, decision trees, or random forests. This makes a direct comparison using the same algorithms and parameters across different tools both complex and often impractical.

By following examples on the AutoKeras website, building and evaluating a basic model was easy, requiring little to no data preprocessing. However, while model customization and tuning are possible, the website resources could be more in-depth and accessible to an inexperienced user. Performance inconsistency between runs was an issue during the analysis. Repeat analyses took different run times and produced models with different accuracies. Run times and accuracies, including means and standard deviations, obtained from three sets of analyses are shown in Table 8.

The model produced from one of these runs (on the unreduced Cleveland dataset, run 3) is shown in Table 9. The resulting model and number of parameters varied widely for the various runs, sometimes having fewer than 1500 parameters and sometimes more than 50,000. Models always consisted of some dense layers and sometimes also included one or more normalization/batch-normalization layers and/or dropout layers. The number of dense layers varied between models, but ReLU activation functions were always used for the hidden layers. The official website documentation for AutoGluon was fairly easy to follow and contained several tutorials on how to use the tool. However, in-depth explanations of how to use and interpret results were lacking and would require the user to have a prior understanding of machine learning. The user would need to look to other resources for a better understanding of machine learning concepts. AutoGluon also tends to default to the best model when displaying results; however, this can be changed with input options. Models with the best validation scores also did not always provide the best accuracy on the test data; however, with more test data, this may have changed and thus more datasets would be needed for testing. Advanced custom metrics are also possible with AutoGluon, but this requires an in-depth understanding of statistics and machine learning that is likely beyond the basic user. A useful feature with AutoGluon is the leaderboard, which displays models based on validation score by default with no inputs or by accuracy if test data are inputted. An example leaderboard of the AutoGluon tool is presented in Table 10.

Table 9. Example neural network produced by AutoKeras.

Layer (Type)	Output Shape	Param
Input$_1$(InputLayer)	[(None, 13)]	0
MultiCategoryEncoding	(None, 13)	
(MultiCategoryEncoding)	0	0
Normalization (Normalization)	(None, 13)	27
Dense (Dense)	0.150825	896
Relu(ReLU)	0.169202	0
Dense$_1$(Dense)	0.223120	16,640
Relu$_1$(ReLU)	0.276816	0
Dense$_2$(Dense)	0.339213	32,896
Relu$_2$(ReLU)	0.414446	0
Dense$_3$(Dense)	0.424510	129
Classificationhead$_1$	0.431894	0
(Activation)	0.460033	
Total params:	50,588	
Trainable params:	50,561	
Non-trainable params:	27	

Table 10. Example AutoGluon leaderboard (unreduced combined dataset).

Model	Score_Test	Score_Val	Pred_Time_Tes	Pred_Time_Val	Fit_Time	Pred_Time_Test_Marginal	Pred_Time_Val_Marginal	Fit_Time_Marginal
0	RandomForestEntr	0.847826	0.797297	0.036896	0.024798	0.2945	0.036896	0.024798
1	WeightedEnsemble_L2	0.847826	0.810811	0.123012	0.074949	0.979291	0.001744	0.000401
2	RandomForestGini	0.842391	0.77027	0.037669	0.025544	0.297175	0.037669	0.025544
3	ExtraTreesGini	0.842391	0.783784	0.041508	0.024531	0.28753	0.041508	0.024531
4	ExtraTreesEntr	0.826087	0.77027	0.046702	0.024206	0.282546	0.046702	0.024206
5	XGBoost	0.809783	0.722973	0.005913	0.003386	0.023183	0.005913	0.003386
6	KNeighborsDist	0.690217	0.668919	0.00367	0.001662	0.004442	0.00367	0.001662
7	KNeighborsUnif	0.690217	0.662162	0.014753	0.002047	0.012023	0.014753	0.002047

PyCaret documentation is well labeled and defined, making it quick to understand the capacity of utility that it has available to offer, covering several machine learning techniques (classification, regression, anomalies, clustering, and time series). Opening any one of the tabs for the desired technique immediately gives insight into the higher workings and functions of the chosen function, which makes it appear convoluted, but those familiar with machine learning will be able to understand all that they provide and the freedom of its utility for these AutoML tools. Functionally, when PyCaret runs with the chosen technique, it provides a leaderboard like other AutoML tools, which displays the top-ranked models by various scores (F1, accuracy, etc.) that best apply to the data that it was given. Across the leaderboard, it usefully highlights the best scores in each given category; however, the ranking will sometimes find better scores for different models as opposed to the best model across the leaderboard. Testing the PyCaret AutoML tool with the Heart Disease datasets was proven to have successful results in identifying efficient models that can be utilized, especially after implementing data preprocessing and reducing the dataset to focus on the parameters that matter more. In some circumstances, there have been less promising results, giving lower accuracy, and this could be because of the differing size of the data in some cases, as well as potentially not reducing the datasets to isolate the more correlated features.

The impact of feature selection performed when reducing the datasets based on correlation was not clear. When using AutoGluon, the accuracy, after reducing for correlation, was worse for the Cleveland dataset, better for the Hungarian, and almost the same for the combined. Results for the unreduced and correlated versions were similar for all three datasets when AutoKeras was used. Interestingly, the accuracy obtained by AutoGluon for the "correlated" Cleveland dataset is among the lowest in Table 4. This could be a case of underfitting for this particular dataset and the model combination. Other than this result, AutoGluon produced consistent results and some of the highest accuracies. The best model, based on validation accuracies, for each analysis was WeightedEnsemble-L2.

Unlike PyCaret and AutoGluon, repeat analyses using AutoKeras produced different models and different accuracies. In addition to the disadvantage of inconsistent results, AutoKeras took the longest to run, with some of its shortest run times still longer than those for the other two tools. The datasets used here were small. The time required by AutoKeras could be a prohibitive problem for large datasets.

Traditional manual steps were applied to create ML models utilizing ten machine learning algorithms and an ensemble model that merged seven well-performing models, and the results are presented in Table 4. When these results are compared with the results achieved using the AutoML tools, it can be noticed that the AutoML tools generated ML models that could perform much better than manually created models. The top-performing models created manually performed with around 60% accuracy, while the top-performing AutoML-generated models performed with around 85% accuracy on the same Cleveland dataset.

In addition to comparing results with those of conventional machine learning models that were generated using sklearn libraries, deep learning models could be manually developed using libraries such as Tensorflow, Keras, or PyTorch to compare with them with the AutoKeras AutoML tool's models. This comparison could include a detailed explanation of parameters and FLOPS used during the training. In the literature, researchers have claimed to achieve up to 95% accuracy on heart disease datasets [12–22,59,60]. However, we did not have a chance to repeat the results of these studies since the code and configurations were not shared. To let other researchers compare their results with ours, we open-sourced our code that utilized AutoML tools and applied traditional model generation steps on GitHub [58]. Accuracies and F1 scores are the key metrics in Tables 5–7; however, accuracy alone should not be considered a true metric as it may not provide a complete picture of a model's performance. It does not consider the nuances of different classes, class imbalances, or the cost associated with misclassification. Relying solely on accuracy may lead to misleading conclusions, especially in scenarios where the dataset is imbalanced or the costs of the false positives and false negatives differ significantly.

When utilizing a machine learning tool to aid in medical diagnosis, it is critical that expert knowledge is employed in both judging the soundness of the medical results and the relevance of a wide range of relevant model metrics. While accuracy measured the overall correctness of the predictions, precision focused on the proportion of true positives among all predicted positives, and recall focused on the proportion of true positives among all actual positives. In scenarios where class imbalance exists, optimizing for high accuracy may result in low precision or recall as the model may favor the majority class. Therefore, a trade-off exists between accuracy and precision/recall, and the choice depends on the specific requirements of the application. F1 scores combine precision and recall into a single metric, providing a balance between the two. However, optimizing for F1 score may not be suitable in all scenarios. In some cases, precision or recall may be more important, and a trade-off exists between F1 score and precision/recall. We decided to utilize accuracy and F1 score metrics since many of the comparable studies utilized these metrics. However, sensitivity and specificity are also important metrics generally used in the medical domain. Further studies should consider utilizing sensitivity and specificity as well. In the process of optimizing a model, it is important to consider that as models become more complex and accurate, they often require more computational resources and longer inference times. There is a trade-off between achieving higher accuracy and the computational cost required for inference. There is also the risk of overfitting. Overfitting occurs when a model performs well on the training data but fails to generalize to unseen data, while underfitting occurs when a model is too simple to capture underlying patterns in the data. These represent opposite ends of a trade-off. Increasing model complexity may reduce underfitting but increase the risk of overfitting, while reducing model complexity may reduce overfitting but increase the risk of underfitting.

In addition to model evaluation metrics, essential to account for are the limitations of the data, assumptions made, and constraints applied. We acknowledge and discuss/explain these uncertainties to help readers gain a more nuanced understanding of the limitations and potential sources of error associated with the evaluation metrics. We hope this will promote transparency and help readers interpret the results more accurately. Dataset limitations included size, quality, representativeness, and potential biases. The datasets used in this paper were very small. Although they have been used in numerous other studies, allowing for easier comparison of results, small size means the datasets are not likely to be representative of large populations and may be more susceptible to selection biases. There were also a large number of missing values in all but the Cleveland datasets. A key assumption made during this analysis was that these datasets and tools can be used to effectively predict heart disease. This is a huge assumption, and it should be noted that employing machine learning, particularly black-box tools like AutoML, requires caution and diligence. The data used here included both male and female patients. However, this may have led to the importance of other features being suppressed. Different features may have different levels of importance for male and female patients as well. This is not accounted for in this paper, but female under-representation in medical research is a well-known and persistent problem. Of the 41,622 participants in US government clinical trials analyzed by Mayor et al., only about 27.5% of them were female. [54] Future research should explore how sex impacts feature importance and model performance. Moreover, larger datasets could be used to assess model performances, including accuracies, precision, recall, and inference times.

Some features were excluded from our analyses using the combined dataset due to missing data. The inclusion of these, as well as other previously unused variables, may have had a significant impact on tool performance. The binary classification was used here; however, all three AutoML tools have the ability to perform multilabel classification, and their performances at this task could be compared in the future. Two previous examples we referenced of AutoML tools being used to analyze heart disease data were from 2021 [8,15].

New studies are needed to comprehensively test the performance of these and additional AutoML tools as these tools continue to be developed and improved. Two AutoML tools that were developed specifically for the medical field were found while working on this project [61–63]. AutoPrognosis was first released in 2018 and uses Bayesian optimization to create and optimize pipeline ensembles [60]. An updated version of the tool, AutoPrognosis 2.0, was released in 2022 [63–65]. Pharm-AutoML was developed by Genentech employees Liu, Lu, and Lu in 2020 [65]. If using csv-formatted data, Pharm-AutoML can handle steps from data preprocessing through model selection and evaluation for multiclassification problems [61]. AutoPrognosis 2.0 and Pharm-AutoML are both open-source and have been used to analyze heart disease datasets, though different from the data we used [62–66].

6. Conclusions

This study ventured into the effectiveness of automated machine learning (AutoML) tools for heart disease diagnosis and discovered that AutoGluon consistently outperformed its peers, with accuracy rates ranging from 78% to 86%. PyCaret's performance was found to vary depending on the dataset, with accuracy rates from 65% to 83%, suggesting a nuanced relationship between tool efficacy and dataset characteristics. AutoKeras showed the most variation in results, with accuracies between 54% and 83%, indicating the potential for high performance but also a significant dependency on the dataset used.

Compared to traditional machine learning methods, which yielded accuracy rates of 55% to 60% via standard practices like exploratory data analysis (EDA), data cleaning, feature engineering, and various modeling algorithms from the sklearn library, the advantage of AutoML tools is clear. This discrepancy illustrates the promising potential of AutoML to revolutionize diagnostic accuracy and make sophisticated analyses more accessible to healthcare practitioners.

The insights from this study suggest that AutoML tools, especially AutoGluon, have the capacity to significantly refine and expedite the diagnostic process, with profound implications for the treatment and prevention of heart disease internationally. Nevertheless, this investigation serves as an initial step towards a broader and more detailed exploration of AutoML's capabilities, underscoring the necessity for future research that includes larger and more varied datasets as well as a wider array of AutoML tools, potentially revolutionizing patient care in the process.

Author Contributions: Conceptualization, Methodology, and Writing—Review and Editing: O.T. and T.C.A.; Investigation, Software, Writing Original Draft: L.M.P., A.H. and A.P.; Validation: L.M.P., O.T. and T.C.A.; Supervision: O.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data are contained within the article.

Acknowledgments: The authors would like to thank the creators of the datasets: Andras Janosi, Hungarian Institute of Cardiology, Budapest; William Steinbrunn, University Hospital, Zurich, Switzerland; Matthias Pfisterer, University Hospital, Basel, Switzerland; and Robert Detrano, V.A. Medical Center, Long Beach and Cleveland Clinic Foundation. We would like to express our sincere gratitude to all who provided insights and constructive feedback, which contributed to the improvement of this article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Gaidai, O.; Cao, Y.; Loginov, S. Global Cardiovascular Diseases Death Rate Prediction. *Curr. Probl. Cardiol.* **2023**, *48*, 101622. [CrossRef]
2. Laslett, L.J.; Alagona, P.; Clark, B.A.; Drozda, J.P.; Saldivar, F.; Wilson, S.R.; Poe, C.; Hart, M. The Worldwide Environment of Cardiovascular Disease: Prevalence, Diagnosis, Therapy, and Policy Issues. *J. Am. Coll. Cardiol.* **2012**, *60*, S1–S49. [CrossRef]
3. Luo, C.; Tong, Y. Comprehensive study and review of coronary artery disease. In Proceedings of the Second International Conference on Biological Engineering and Medical Science (ICBioMed 2022), Oxford, UK, 7–13 November 2022. [CrossRef]

4. Absar, N.; Das, E.K.; Shoma, S.N.; Khandaker, M.U.; Miraz, M.H.; Faruque, M.R.I.; Tamam, N.; Sulieman, A.; Pathan, R.K. The Efficacy of Machine-Learning-Supported Smart System for Heart Disease Prediction. *Healthcare* **2022**, *10*, 1137. [CrossRef]
5. Rani, U. Analysis of Heart Diseases Dataset Using Neural Network Approach. *Int. J. Data Min. Knowl. Manag. Process* **2011**, *1*, 1–8. [CrossRef]
6. Singh, P.; Singh, S.; Pandi-Jain, G.S. Effective heart disease prediction system using data mining techniques. *Int. J. Nanomed.* **2018**, *13*, 121–124. [CrossRef]
7. Ismail, A.; Ravipati, S.; Gonzalez-Hernandez, D.; Mahmood, H.; Imran, A.; Munoz, E.J.; Naeem, S.; Abdin, Z.U.; Siddiqui, H.F. Carotid Artery Stenosis: A Look into the Diagnostic and Management Strategies, and Related Complications. *Cureus* **2023**, *15*, e38794. [CrossRef]
8. Pol, U.R.; Sawant, T.U. Automl: Building a classification model with PyCaret. *YMER* **2021**, *20*, 547–552. [CrossRef]
9. Ferreira, L.; Pilastri, A.; Martins, C.M.; Pires, P.M.; Cortez, P. A Comparison of AutoML Tools for Machine Learning, Deep Learning and XGBoost. In Proceedings of the 2021 International Joint Conference on Neural Networks (IJCNN), Shenzhen, China, 18–22 July 2021; pp. 1–8. [CrossRef]
10. Lenkala, S.; Marry, R.; Gopovaram, S.R.; Akinci, T.C.; Topsakal, O. Comparison of Automated Machine Learning (AutoML) Tools for Epileptic Seizure Detection Using Electroencephalograms (EEG). *Computers* **2023**, *12*, 197. [CrossRef]
11. Topsakal, O.; Akinci, T.C. Classification and Regression Using Automatic Machine Learning (AutoML)–Open Source Code for Quick Adaptation and Comparison. *Balk. J. Electr. Comput. Eng.* **2023**, *11*, 257–261. [CrossRef]
12. Hazra, A.; Mandal, S.K.; Gupta, A.; Mukherjee, A.; Mukherjee, A. Heart disease diagnosis and prediction using machine learning and data mining techniques: A review. *Adv. Comput. Sci. Technol.* **2017**, *10*, 2137–2159.
13. Khan, Y.; Qamar, U.; Yousaf, N.; Khan, A. Machine learning techniques for heart disease datasets: A survey. In Proceedings of the 2019 11th International Conference on Machine Learning and Computing (ICMLC '19), Zhuhai, China, 22–24 February 2019; ACM: New York, NY, USA, 2019; pp. 27–35. [CrossRef]
14. Marimuthi, M.; Abinaya, M.; Hariesh, K.S.; Madhankumar, K.; Pavithra, V. A review on heart disease prediction using machine learning and data analytics approach. *Int. J. Comput. Appl.* **2018**, *181*, 20–25. [CrossRef]
15. Nagavelli, U.; Samanta, D.; Chakraborty, P. Machine Learning Technology-Based Heart Disease Detection Models. *J. Healthc. Eng.* **2022**, *2022*, 7351061. [CrossRef] [PubMed]
16. Li, Y.; Shen, F.; Hu, L.; Lang, Z.; Liu, L.D.; Cai, F.; Fu, L. A Stare-Down Video-Rate High-Throughput Hyperspectral Imaging System and Its Applications in Biological Sample Sensing. *IEEE Sens. J.* **2023**, *23*, 23629–23637. [CrossRef]
17. Shen, F.; Deng, H.; Yu, L.; Cai, F. Open-source mobile multispectral imaging system and its applications in biological sample sensing. *Spectrochim. Acta Part A Mol. Biomol. Spectrosc.* **2022**, *280*, 121504. [CrossRef] [PubMed]
18. Squiers, J.J.; Thatcher, J.E.; Bastawros, D.S.; Applewhite, A.J.; Baxter, R.D.; Yi, F.; Quan, P.; Yu, S.; DiMaio, J.M.; Gable, D.R. Machine learning analysis of multispectral imaging and clinical risk factors to predict amputation wound healing. *J. Vasc. Surg.* **2022**, *75*, 279–285. [CrossRef]
19. Staszak, K.; Tylkowski, B.; Staszak, M. From Data to Diagnosis: How Machine Learning Is Changing Heart Health Monitoring. *Int. J. Environ. Res. Public Health* **2023**, *20*, 4605. [CrossRef]
20. Padmanabhan, M.; Yuan, P.; Chada, G.; Nguyen, H.V. Physician-friendly machine learning: A case study with cardiovascular disease risk prediction. *J. Clin. Med.* **2019**, *8*, 1050. [CrossRef]
21. Valarmathi, R.; Sheela, T. Heart disease prediction using hyperparameter optimization (HPO) tuning. *Biomed. Signal Process. Control* 2021. [CrossRef]
22. Romero, R.A.A.; Deypalan, M.N.Y.; Mehrotra, S.; Jungao, J.T.; Sheils, N.E.; Manduchi, E. Benchmarking AutoML frameworks for disease prediction using medical claims. *BioData Min.* **2022**, *15*, 15. [CrossRef]
23. Wang, X.; Zhang, Z.; Zhu, W. Automated graph machine learning: Approaches, libraries, and directions. *arXiv* **2022**, arXiv:2201.01288. [CrossRef]
24. Bu, C.; Lu, Y.; Liu, F. Automatic Graph Learning with Evolutionary Algorithms: An Experimental Study. In *PRICAI 2021: Trends in Artificial Intelligence. PRICAI 2021, Hanoi, Vietnam, 8–12 November 2021*; Pham, D.N., Theeramunkong, T., Governatori, G., Liu, F., Eds.; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2021; Volume 13031. [CrossRef]
25. Alamin, M.A. Democratizing Software Development and Machine Learning Using Low Code Applications. Master's Thesis, University of Calgary, Calgary, AB, Canada, 2022.
26. Topsakal, O.; Dobratz, E.J.; Akbas, M.I.; Dougherty, W.M.; Akinci, T.C.; Celikoyar, M.M. Utilization of Machine Learning for the Objective Assessment of Rhinoplasty Outcomes. *IEEE Access* **2023**, *11*, 42135–42145. [CrossRef]
27. Madhugiri, D. Beginner's Guide to AutoML with an Easy AutoGluon Example. Analytics Vidhya, 18 September 2022. Available online: https://www.analyticsvidhya.com/blog/2021/10/beginners-guide-to-automl-with-an-easy-autogluon-example/ (accessed on 9 September 2023).
28. Jin, H.; Chollet, F.; Song, Q.; Hu, X. AutoKeras: An AutoML Library for Deep Learning. *J. Mach. Learn. Res.* **2023**, *24*, 1–6.
29. Budjac, R.; Nikmon, M.; Schreiber, P.; Zahradnikova, B.; Janacova, D. Automated machine learning overview. *Sciendo* **2019**, *27*, 107–112. [CrossRef]
30. Koh, J.C.O.; Spangenberg, G.; Kant, S. Automated Machine Learning for High-Throughput Image-Based Plant Phenotyping. *Remote Sens.* **2021**, *13*, 858. [CrossRef]

31. Singh, V.K.; Josh, K. Automated Machine Learning (AutoML): An overview of opportunities for application and research. *J. Inf. Technol. Case Appl. Res.* **2022**, *24*, 75–85. [CrossRef]
32. Lee, S.; Kim, J.; Bae, J.H.; Lee, G.; Yang, D.; Hong, J.; Lim, K.J. Development of Multi-Inflow Prediction Ensemble Model Based on Auto-Sklearn Using Combined Approach: Case Study of Soyang River Dam. *Hydrology* **2023**, *10*, 90. [CrossRef]
33. Pushparaj, S.N.; Sivasankaran, S.M.; Thamizh Chem-mal, S. Prediction of Heart Disease Using a Hybrid of CNN-LSTM Algorithm. *J. Surv. Fish. Sci.* **2023**, *10*, 5700–5710.
34. Ferreira, L.; Pilastri, A.L.; Henrique, C.; Santos, P.A.; Cortez, P. A Scalable and Automated Machine Learning Framework to Support Risk Management. *Lect. Notes Comput. Sci.* **2020**, *12613*, 291–307. [CrossRef]
35. Egger, R. Machine Learning in Tourism: A Brief Overview. In *Applied Data Science in Tourism*; Spring: Berlin/Heidelberg, Germany, 2022. [CrossRef]
36. Yang, S.; Bhattacharjee, D.; Kumar, V.B.Y.; Chatterjee, S.; De, S.; Debacker, P.; Verkest, D.; Mallik, A.; Catthoor, F. AERO: Design Space Exploration Framework for Resource-Constrained CNN Mapping on Tile-Based Accelerators. *IEEE J. Emerg. Sel. Top. Circuits Syst.* **2022**, *12*, 508–521. [CrossRef]
37. Sarangpure, N.; Dhamde, V.; Roge, A.; Doye, J.; Patle, S.; Tamboli, S. Automating the Machine Learning Process using PyCaret and Streamlit. In Proceedings of the 2023 2nd International Conference for Innovation in Technology (INOCON), Bangalore, India, 3–5 March 2023; pp. 1–5. [CrossRef]
38. Vinicius, M.; Paulo, N.; Cecilia, M. Auto machine learning to predict pregnancy after fresh embryo transfer following in vitro fertilization. *World J. Adv. Res. Rev.* **2022**, *16*, 621–626. [CrossRef]
39. Olson, R.S. TPOT. Available online: http://epistasislab.github.io/tpot/ (accessed on 3 March 2023).
40. Gurdo, N.; Volke, D.C.; McCloskey, D.; Nikel, P.I. Automating the design-build-test-learn cycle towards next-generation bacterial cell factories. *New Biotechnol.* **2023**, *74*, 1–15. [CrossRef] [PubMed]
41. Erickson, N.; Mueller, J.; Shirkov, A.; Zhang, H.; Larroy, P.; Li, M.; Smola, A. AutoGluon-Tabular: Robust and Accurate AutoML for Structured Data. *arXiv* **2020**, arXiv:2003.06505.
42. Ali, A.A.; Khedr, A.M.; El-Bannany, M.; Kanakkayil, S. A Powerful Predicting Model for Financial Statement Fraud Based on Optimized XGBoost Ensemble Learning Technique. *Appl. Sci.* **2023**, *13*, 2272. [CrossRef]
43. Gaur, S.; Kalani, P.; Mohan, M. Harmonic-to-noise ratio as a speech biomarker for fatigue: K-nearest neighbour machine learning algorithm. *Med. J. Armed Forces India* **2023**. [CrossRef]
44. Jawad, B.J.; Shaker, S.M.; Altintas, I.; Eugen-Olse, J.; Nehlin, J.; Andersen, O.; Kallemose, T. Development and validation of prognostic machine learning models for short- and long-term mortality among acutely hospitalized patients. *Eur. PMC* **2023**. [CrossRef]
45. Suresh, K.; Elkahwagi, M.A.; Garcia, A.; Naples, J.G.; Corrales, C.E.; Crowson, M.G. Development of a Predictive Model for Persistent Dizziness Following Vestibular Schwannoma Surgery. *Laryngoscope* **2023**, *133*, 3534–3539. [CrossRef] [PubMed]
46. Ortiz-Perez, A.; Izquierdo Lozano, C.; Meijers, R.; Grisoni, F.; Albertazzi, L. Identification of fluorescently-barcoded nanoparticles using machine learning. *Nanoscale Adv.* **2023**, *5*, 2307–2317. [CrossRef]
47. Ehlers, M.R.; Lonsdorf, T.B. Data sharing in experimental fear and anxiety research: From challenges to a dynamically growing database in 10 simple steps. *Neurosci. Biobehav. Rev.* **2022**, *143*, 104958. [CrossRef]
48. Lu, P.J.; Chuang, J.-H. Fusion of Multi-Intensity Image for Deep Learning-Based Human and Face Detection. *IEEE Access* **2022**, *10*, 8816–8823. [CrossRef]
49. Maghfour, J.; Ceresnie, M.; Olson, J.; Lim, H.W. The association between frontal fibrosing alopecia, sunscreen, and moisturizers: A systematic review and meta-analysis. *J. Am. Acad. Dermatol.* **2022**, *87*, 395–396. [CrossRef]
50. Datasets | Kaggle. Kaggle.com. 2019. Available online: https://www.kaggle.com/datasets (accessed on 25 April 2023).
51. UCI Machine Learning Repository: Data Sets. Uci.edu. 2009. Available online: https://archive.ics.uci.edu/dataset/45/heart+disease (accessed on 18 April 2023).
52. Price, W.N., II; Cohen, I.G. Privacy in the age of medical big data. *Nat. Med.* **2019**, *25*, 37–43. [CrossRef] [PubMed]
53. Cleveland, Hungarian, Switzerland, and VA Datasets. Available online: https://archive.ics.uci.edu/ml/datasets/heart+disease (accessed on 9 September 2023).
54. Pathare, A.; Mangrulkar, R.; Suvarna, K.; Parekh, A.; Thakur, G.; Gawade, A. Comparison of tabular synthetic data generation techniques using propensity and cluster log metric. *Int. J. Inf. Manag. Data Insights* **2023**, *3*, 100177. [CrossRef]
55. El-Bialy, R.; Salamay, M.A.; Karam, O.H.; Khalifa, M.E. Feature analysis of coronary artery heart disease data sets. *Procedia Comput. Sci.* **2015**, *65*, 459–468. [CrossRef]
56. Sarra, R.R.; Dinar, A.M.; Mohammed, M.A.; Abdulkareem, K.H. Enhanced heart diseaseprediction based on machine learning and X2 statistical optimal feature selection model. *Designs* **2022**, *6*, 87. [CrossRef]
57. Ahmed, I. A Study of Heart Disease Diagnosis Using Machine Learning and Data Mining. Master's Thesis, California State University, San Bernardino, CA, USA, 2022. Volume 1591. Available online: https://scholarworks.lib.csusb.edu/etd/1591 (accessed on 9 September 2023).
58. AutoML Comparison for Heart Disease Diagnosis GitHub Page. Available online: https://github.com/researchoutcome/automl-comparison-heart/ (accessed on 4 July 2023).
59. Chandrasekhar, N.; Peddakrishna, S. Enhancing Heart Disease Prediction Accuracy through Machine Learning Techniques and Optimization. *Processes* **2023**, *11*, 1210. [CrossRef]

60. Mayor, J.M.; Preventza, O.; McGinigle, K.; Mills, J.L.; Montero-Baker, M.; Gilani, R.; Pallister, Z.; Chung, J. Persistent underrepresentation of female patients in United States trials of common vascular diseases from 2008 to 2020. *J. Vasc. Surg.* **2022**, *75*, 30–36. [CrossRef] [PubMed]
61. Finkelhor, R.S.; Newhouse, K.E.; Vrobel, T.R.; Miron, S.D.; Bahler, R.C. The ST segment/heartrate slope as a predictor of coronary artery disease: Comparison with quantitative thallium imaging and conventional ST segment criteria. *Am. Heart J.* **1986**, *112*, 296–304. [CrossRef]
62. Islam, M.M.; Haque, M.R.; Iqbal, H.; Hasan, H.M.M.; Hasan, M.; Kabir, M.N. Breast cancer prediction: A comparative study using machine learning techniques. *SN Comput. Sci.* **2020**, *1*, 290. [CrossRef]
63. Alaa, A.M.; van der Schaar, M. AutoPrognosis: Automated clinical prognostic modeling via Bayesian optimization with structured kernel learning. *arXiv* **2018**, arXiv:1802.07207. [CrossRef]
64. Imrie, F.; Cebere, B.; McKinney, E.F.; van der Schaar, M. AutoPrognosis 2.0: Democratizing diagnostic and prognostic modeling in healthcare with automated machine learning. *arXiv* **2022**, arXiv:2210.12090. [CrossRef]
65. Liu, G.; Lu, D.; Lu, J. Pharm-AutoML: An open-source, end-to-end automated machine learning package for clinical outcome prediction. *CPT Pharmacomet. Syst. Pharmacol.* **2021**, *10*, 478–488. [CrossRef] [PubMed]
66. Alaa, A.M.; van der Schaar, M. Cardiovascular disease risk prediction using automated machine learning: A prospective study of 423,604 UK Biobank participants. *PLoS ONE* **2019**, *14*, e0213653. [CrossRef] [PubMed]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.