Report on AUTO-ML experiments

Student Name: Provost Simon.
Professor Name: Alex A. Freitas.

Jul 2021

1 Introduction

The following experiment demonstrates how the Auto-ML (i.e.: Auto-Sklearn) framework handles classification of medical datasets across a variety of different dimensions in the field of Healthcare. The datasets and their caracteristics used for the experiments are available in the section 2. However, to be consistent each run was performed on a Google cloud compute engine virtual machine instance with the following characteristics:

Table 1: Google Cloud virtual machine instance characteristics

Param name	**Param value**
Machine type	e2-standard-2
CPU	2 vCPU
Virtual Machine Memory	8 Go
Image size	10 Go

2 Datasets characteristics

The datasets used in the experiments that follow are described in detail in the table 2. (1) The dataset name indicates the dataset's name; (2) The number of samples indicates the total number of samples available for this dataset; (3) The number of features indicates the number of features available without the class labels column; and (4) The number of classes indicates the number of distinct class labels associated with the appropriate dataset, which also indicates whether it is a binary/multi class classification problem. Finally, (5) the column class_label x indicates the proportion of samples for that class in the entire dataset.

Here are the datasets source:

- Thoracic Surgery dataset [1].
- Diabetic Retinopathy Debrecen dataset [2].
- Estimation of obesity levels based on eating habits and physical condition [3].
- Breast Cancer Coimbra Data Set [4].
- Heart failure clinical records Data Set [5].

Table 2: Datasets used for the experiments.

dataset name	number of samples	number of features	number of classes	class_label_1(%)	class_label_2(%)	class_label_3(%)	class_label_4(%)	class_label_5(%)	class_label_6(%)	class_label_7(%)
Breast-cancer-Coimbra	116	9	2	44.827586	55.172414					
diabetic-retinopathy-Debrecen	1151	19	2	46.915725	53.084275					
Heart-failure-clinical-records_dataset	299	12	2	67.892977	32.107023					
ObesityDataSet_raw_and_data_synthetic	2111	16	7	12.884889	13.595452	16.627191	14.069162	15.348176	13.737565	13.737565
Thoracic-Surgery-binary-survival	470	16	2	85.106383	14.893617					

3 Auto-Sklearn characteristics

The following table depicts the Auto-Sklearn's parameters used for the experiments (refer to the Auto-Sklearn API for more details):

Table 3: Auto-Sklearn parameters

Param name	Param value
time_left_for_this_task	3600
per_run_time_limit	360
n_jobs	4 (i.e.: maximum 4, if only 2 available 2 will be taken)
memory_limit	5000
seed	2/11/21/42/85
resampling_strategy	holdout
ensemble_size	50

4 Auto-Weka characteristics

The following table depicts the Auto-Weka's parameters used for the experiments (refer to the Auto-Weka API for more details):

Table 4: Auto-Weka parameters

Param name	**Param value**
-t <name of training file>	training_set of the fold (i)
-T <name of test file>	test_set of the fold (i)
-timeLimit <limit> (approximately in minutes)	60 (3600 seconds)
-seed <seed>	85
-memLimit <limit> (in MiB)	5000
-nBestConfigs <limit>	50

5 Phase 1 results of the experiment

5.1 Experiment details

The following experiment uses Auto-Sklearn to search for the best model to classify an entire dataset given.

For the results available in the table 5, Auto-Sklearn[6] (i.e.: An Auto-ML Framework) was applied to the entire dataset chosen. This relies on Auto-ML doing an internal partition of the data into training and validation during its run.

5.2 Table columns details

Each row shows a set of different metrics: (1) The dataset's name; (2) the classifier selected by the AutoML pipeline; (3 & 4) the search time limit in seconds as well as the duration of the algorithm; (5) the seed value, which ensures reproducibility of the model if the results are consistent across different seeds (for seeds chosen see section 3); (6) the accuracy score of this classifier on the test data previously determined by the initial data split at the start of the pipeline; (7 & 8) The recall& precision score which is the ability of the classifier to find all the positive samples; (9 & 10) The F1 score macro and micro are used which depicts for macro the following: compute te metric independently for each class and thus to compute an average that compares all classes equally, and for micro: aggregates the contributions of all classes to compute the average metric; Finally, (11) the AUROC (i.e: receiver operating characteristic curve) score, which is a performance metric for discrimination: it tells us about the model's ability to discriminate between cases - positive examples - and non-cases - negative examples - (i.e.: Higher is the score better is the classifier).

Table 5: Optimum classifiers chosen from the Auto-Sklearn pipeline on the datasets presented section 2. Hyperparameters are available in the appendix (see HP column). Note: HP stands for Hyper Parameters.

Dataset name	HP	Classifier	Search Time limit	Algorithm time run (s)	Seed	Accuracy	Precision	Recall	F1 score macro	F1 score micro	Auroc
breast-cancer	Breast-1	mlp	3600	2.482066	11	0.827586	0.825758	0.816176	0.819876	0.827586	0.816176
breast-cancer	Breast-2	qda	3600	3.506922	2	0.517241	0.558333	0.551471	0.512019	0.517241	0.551471
breast-cancer	Breast-3	mlp	3600	3.074595	21	0.793103	0.798077	0.795238	0.792857	0.793103	0.795238
breast-cancer	Breast-4	random_forest	3600	2.54832	42	0.793103	0.798077	0.795238	0.792857	0.793103	0.795238
breast-cancer	Breast-5	random_forest	3600	6.722548	85	0.655172	0.637019	0.658333	0.633838	0.655172	0.658333
Heart-failure	Heart-failure-1	libsvm_svc	3600	2.202007	11	0.92	0.919118	0.902669	0.91	0.92	0.902669
Heart-failure	Heart-failure-2	mlp	3600	4.372969	2	0.866667	0.831104	0.849206	0.839194	0.866667	0.849206
Heart-failure	Heart-failure-3	k_nearest_neighbors	3600	2.542999	21	0.826667	0.766667	0.733918	0.747475	0.826667	0.733918
Heart-failure	Heart-failure-4	random_forest	3600	4.143286	42	0.746667	0.738889	0.736437	0.737521	0.746667	0.736437
Heart-failure	Heart-failure-5	k_nearest_neighbors	3600	3.839199	85	0.8	0.716374	0.725152	0.720497	0.8	0.725152
Obesity	obesity-1	gradient_boosting	3600	22.408372	11	0.965909	0.966143	0.96609	0.965447	0.965909	0.999414
Obesity	obesity-2	gradient_boosting	3600	131.172882	2	0.982955	0.98177	0.98325	0.982323	0.982955	0.999707
Obesity	obesity-3	gradient_boosting	3600	7.056944	21	0.971591	0.969771	0.969468	0.969479	0.971591	0.998308
Obesity	obesity-4	gradient_boosting	3600	28.593236	42	0.960227	0.958619	0.960982	0.959452	0.960227	0.99933
Obesity	obesity-5	gradient_boosting	3600	9.831563	85	0.982955	0.981575	0.982844	0.982134	0.982955	0.999591
Thoracic-Surgery	Thoracic-Surgery-1	sgd	3600	1.728338	11	0.79661	0.401709	0.494737	0.443396	0.79661	0.494737
Thoracic-Surgery	Thoracic-Surgery-2	bernoulli_nb	3600	2.939708	2	0.847458	0.438596	0.480769	0.458716	0.847458	0.480769
Thoracic-Surgery	Thoracic-Surgery-3	bernoulli_nb	3600	2.434407	21	0.779661	0.562319	0.509247	0.473214	0.779661	0.509247
Thoracic-Surgery	Thoracic-Surgery-4	gradient_boosting	3600	3.708382	42	0.79661	0.571014	0.511213	0.481319	0.79661	0.511213
Thoracic-Surgery	Thoracic-Surgery-5	decision_tree	3600	3.101079	85	0.830508	0.588406	0.516215	0.498726	0.830508	0.516215
Diabetic-retinopathy	diabetic-retinopathy-1	mlp	3600	13.145924	11	0.704861	0.704915	0.704371	0.70443	0.704861	0.704371
Diabetic-retinopathy	diabetic-retinopathy-2	random_forest	3600	13.08779	2	0.715278	0.718083	0.718083	0.715278	0.715278	0.718083
Diabetic-retinopathy	diabetic-retinopathy-3	libsvm_svc	3600	3.151908	21	0.784722	0.785642	0.787194	0.784556	0.784722	0.787194
Diabetic-retinopathy	diabetic-retinopathy-4	mlp	3600	7.289334	42	0.760417	0.760417	0.762557	0.759928	0.760417	0.762557
Diabetic-retinopathy	diabetic-retinopathy-5	mlp	3600	4.642596	85	0.767361	0.77019	0.76804	0.767021	0.767361	0.76804

6 Phase 2 results of the experiment

6.1 Experiment details

The following experiment perform a 10-fold cross validation of the dataset using a hand-made python program and Auto-Sklearn.

For the results table 6 7 8 9 10 Auto-Sklearn[6] (i.e.: An Auto-ML Framework) and Scikit Learn (i.e.: A Machine learning framework) was applied to follow the experiment. A Python program has been programmed (see pseudo-code section 6.1.1) in order to follow a 10-fold cross-validation process. The program's steps are as follows: (1) The program splits the dataset randomly into ten-folds; (2) drops one fold for later use; (3) uses the remaining folds to feed the Auto-ML pipeline to find the best model using this training samples; (4) once Auto-Sklearn output the best fitted model, the fold dropped previously becomes useful because Sci-kit-learn will use it to determine the viability of the best fitted model on this "unseen" sample data; and (5) this process is repeated ten times, with each time adding another fold for the test_set (i.e.: unseen data), and the remains one feeding Auto-Sklearn to generate another fitted best model on that data. Finally, as a result, the program would output ten models with ten results on unseen data for one dataset. Additonally, via the help of bash script, this entire process is then repeated for each dataset, resulting in the tables shown below.

6.1.1 Pseudo-code of the Python Program

Algorithm 1 Three main steps of the 10-fold cross-validation process. *Note: See snippets belows for more details.*

```
k_folds_split(k_folds ← 10)
cross_validation_process(all_10_folds)
show_latex_cross_validation_process()
```

Algorithm 2 Function that split the dataset into 10 random folds.

```
function K_FOLDS_SPLIT(entireDataset, k_folds)
    data ← entireDataset.copy()
    data ← data.sample(frac ← 1)
    all_10_folds ← array_split(data, k_folds)
end function
```

Algorithm 3 Function that do the cross-validation process over the 10 random folds. *Note: The reSampling if condition determines whether a dataset needs to be re sampled due to inconsistency in its data. At the moment, the feature has been used only once and does not result in something better, the feature remains still available.*

```
1: function CROSS_VALIDATION_PROCESS(all_10_folds, reSampling ← False)
2:
3:     for idx, fold ∈ enumerate(all_10_folds) do
4:         training_set ← all_10_folds.copy()
5:         training_set.pop(idx)
6:         test_set ← fold
7:
8:         input, output ← __get_inputs_outputs_from_folds(training_set)
9:         if reSampling then
10:            input, output ← reSamplingSMOTE(training_set, input, output)
11:        end if
12:        setup(input, output)
13:        fit_predict()
14:
15:        save_model(path ← filepath)
16:        loaded_classifier ← load_model(path ← filepath)
17:        all_best_models.append(loaded_classifier)
18:
19:        crossValUnseenDataOutput ← __predict_unseen_data_cross_validation_pro-
   cess(test_set, loaded_classifier)
20:        ensemble_results.append(crossValUnseenDataOutput[0])
21:    end for
22: end function
```

Note: The following tables will be updated as soon as the Google Cloud virtual machine instance's running analysis is completed. (If you are currently reading a local file, check the following link sometimes to see if a new version is available: link.) - A new analysis is currently running because of the fact that we did change a little bit the metrics evaluation.

Table 6: Diabetic-retinopathy-Debrecen

Fold	HP	Classifier	Search Time (s)	Algorithm run (s)	Seed	Accuracy	recall	F1 macro	F1 micro	AUROC
1	diabetic-fold-1	mlp	3600	7.430558	85	0.75	0.757212	0.749981	0.75	0.757212
2	diabetic-fold-2	random_forest	3600	4.725435	85	0.730435	0.729545	0.729699	0.730435	0.729545
3	diabetic-fold-3	mlp	3600	17.021097	85	0.756522	0.752429	0.75337	0.756522	0.752429
4	diabetic-fold-4	libsvm_svc	3600	2.086872	85	0.704348	0.704311	0.703788	0.704348	0.704311
5	diabetic-fold-5	passive_aggressive	3600	2.18941	85	0.73913	0.735606	0.735754	0.73913	0.735606
6	diabetic-fold-6	liblinear_svc	3600	1.766577	85	0.791304	0.799231	0.790909	0.791304	0.799231
7	diabetic-fold-7	mlp	3600	19.064871	85	0.721739	0.721996	0.716923	0.721739	0.721996
8	diabetic-fold-8	mlp	3600	10.363527	85	0.756522	0.764286	0.752308	0.756522	0.764286
9	diabetic-fold-9	lda	3600	2.402991	85	0.713043	0.718807	0.712957	0.713043	0.718807
10	diabetic-fold-10	mlp	3600	4.246183	85	0.704348	0.703721	0.702526	0.704348	0.703721
Average	N/A	N/A	N/A	7.130	85.000	0.737	0.739	0.735	0.737	0.739

HP stands for Hyper parameter.
The metrics available here are the results of running the fitted model on unseen data using Scikit-Learn.

Table 7: Obesity Dataset raw and data syntheti

Fold	HP	Classifier	Search Time (s)	Algorithm run (s)	Seed	Accuracy	recall	F1 macro	F1 micro	AUROC
1	Obesity-fold-1	gradient_boosting	3600	20.688436	85	0.95283	0.951835	0.951791	0.95283	0.996496
2	Obesity-fold-2	libsvm_svc	3600	4.098679	85	0.971564	0.971001	0.971814	0.971564	0.997796
3	Obesity-fold-3	libsvm_svc	3600	3.354564	85	0.981043	0.978807	0.979618	0.981043	0.999346
4	Obesity-fold-4	gradient_boosting	3600	86.799546	85	0.943128	0.938914	0.932557	0.943128	0.996662
5	Obesity-fold-5	gradient_boosting	3600	170.421677	85	0.966825	0.968942	0.969339	0.966825	0.999287
6	Obesity-fold-6	gradient_boosting	3600	28.833921	85	0.971564	0.970605	0.971264	0.971564	0.998435
7	Obesity-fold-7	gradient_boosting	3600	13.595027	85	0.971564	0.967386	0.968472	0.971564	0.999332
8	Obesity-fold-8	gradient_boosting	3600	125.910552	85	0.957346	0.955436	0.954633	0.957346	0.997772
9	Obesity-fold-9	libsvm_svc	3600	3.559975	85	0.981043	0.976264	0.978697	0.981043	0.999612
10	Obesity-fold-10	libsvm_svc	3600	3.421645	85	0.976303	0.974779	0.976184	0.976303	0.999324
Average	N/A	N/A	N/A	46.068	85.000	0.967	0.965	0.965	0.967	0.998

HP stands for Hyper parameter.
The metrics available here are the results of running the fitted model on unseen data using Scikit-Learn.

Table 8: Thoracic Surgery Binary Survival

Fold	HP	Classifier	Search Time (s)	Algorithm run (s)	Seed	Accuracy	recall	F1 macro	F1 micro	AUROC
1	Thoracic-surgery-fold-1	lda	3600	3.810117	85	0.893617	0.5	0.47191	0.893617	0.5
2	Thoracic-surgery-fold-2	random_forest	3600	6.874774	85	0.87234	0.5	0.465909	0.87234	0.5
3	Thoracic-surgery-fold-3	random_forest	3600	6.621401	85	0.808511	0.5	0.447059	0.808511	0.5
4	Thoracic-surgery-fold-4	lda	3600	1.946631	85	0.893617	0.5	0.47191	0.893617	0.5
5	Thoracic-surgery-fold-5	adaboost	3600	6.081775	85	0.893617	0.488372	0.47191	0.893617	0.488372
6	Thoracic-surgery-fold-6	k_nearest_neighbors	3600	1.577444	85	0.87234	0.5	0.465909	0.87234	0.5
7	Thoracic-surgery-fold-7	adaboost	3600	6.760939	85	0.893617	0.5	0.47191	0.893617	0.5
8	Thoracic-surgery-fold-8	k_nearest_neighbors	3600	2.219199	85	0.765957	0.486486	0.433735	0.765957	0.486486
9	Thoracic-surgery-fold-9	extra_trees	3600	5.944518	85	0.702128	0.49881	0.472756	0.702128	0.49881
10	Thoracic-surgery-fold-10	lda	3600	2.401455	85	0.808511	0.487179	0.447059	0.808511	0.487179
Average	N/A	N/A	N/A	4.42	85.00	0.84	0.50	0.46	0.84	0.50

HP stands for Hyper parameter. (Note: Hyper parameter will be available during the last analysis of the phase
The metrics available here are the results of running the fitted model on unseen data using Scikit-Learn. Note:

Table 9: Breast cancer Coimbra

" Fold	HP	Classifier	Search Time (s)	Algorithm run (s)	Seed	Accuracy	recall	F1 macro	F1 micro	AUROC
1	Breast-cancer-fold-1	mlp	3600	2.962654	85	0.833333	0.875	0.828571	0.833333	0.875
2	Breast-cancer-fold-2	mlp	3600	3.331546	85	0.75	0.75	0.733333	0.75	0.75
3	Breast-cancer-fold-3	mlp	3600	4.40299	85	1.0	1.0	1.0	1.0	1.0
4	Breast-cancer-fold-4	gradient_boosting	3600	3.942139	85	0.666667	0.657143	0.657143	0.666667	0.657143
5	Breast-cancer-fold-5	lda	3600	2.19195	85	0.833333	0.857143	0.833333	0.833333	0.857143
6	Breast-cancer-fold-6	bernoulli_nb	3600	1.271993	85	0.833333	0.833333	0.828571	0.833333	0.833333
7	Breast-cancer-fold-7	lda	3600	1.556687	85	0.545455	0.482143	0.47619	0.545455	0.482143
8	Breast-cancer-fold-8	adaboost	3600	3.883168	85	1.0	1.0	1.0	1.0	1.0
9	Breast-cancer-fold-9	mlp	3600	3.241668	85	0.727273	0.716667	0.717949	0.727273	0.716667
10	Breast-cancer-fold-10	sgd	3600	1.50605	85	0.818182	0.5	0.45	0.818182	0.5
Average	N/A	N/A	N/A	2.83	85.00	0.80	0.77	0.75	0.80	0.77

HP stands for Hyper parameter. (Note: Hyper parameter will be available during the last analysis of the phase
The metrics available here are the results of running the fitted model on unseen data using Scikit-Learn.

Table 10: Heart failure clinical records

Fold	HP	Classifier	Search Time (s)	Algorithm run (s)	Seed	Accuracy	recall	F1 macro	F1 micro	AUROC
1	Heart-failure-fold-1	random_forest	3600	3.684226	85	0.9	0.9	0.89011	0.9	0.9
2	Heart-failure-fold-2	extra_trees	3600	5.542089	85	0.866667	0.856459	0.856459	0.866667	0.856459
3	Heart-failure-fold-3	extra_trees	3600	2.990865	85	0.8	0.75	0.761905	0.8	0.75
4	Heart-failure-fold-4	gradient_boosting	3600	2.314326	85	0.766667	0.761364	0.730424	0.766667	0.761364
5	Heart-failure-fold-5	random_forest	3600	3.395769	85	0.8	0.805556	0.79638	0.8	0.805556
6	Heart-failure-fold-6	random_forest	3600	2.893426	85	0.866667	0.861607	0.864253	0.866667	0.861607
7	Heart-failure-fold-7	random_forest	3600	3.739588	85	0.8	0.770186	0.744318	0.8	0.770186
8	Heart-failure-fold-8	extra_trees	3600	3.429965	85	0.866667	0.464286	0.464286	0.866667	0.464286
9	Heart-failure-fold-9	random_forest	3600	2.849895	85	0.933333	0.920635	0.920635	0.933333	0.920635
10	Heart-failure-fold-10	random_forest	3600	3.327129	85	0.862069	0.835859	0.847368	0.862069	0.835859
Average	N/A	N/A	N/A	3.42	85.00	0.85	0.79	0.79	0.85	0.79

HP stands for Hyper parameter. (Note: Hyper parameter will be available during the last analysis of the phase
The metrics available here are the results of running the fitted model on unseen data using Scikit-Learn.

7 Phase 3 results of the experiment

7.1 Experiment details

The following experiment perform a 10-fold cross validation of the dataset using a hand-made python program and Auto-Weka.

To follow the experiment and produce the results table 11 12 13 14 15, Auto-Weka (i.e., an Auto-ML Framework) was applied. To follow a 10-fold cross-validation process, the python program showed section 6.1.1 was refactored (i.e, The refactored python program is available section 7.1.1). The steps of the new program are as follows: (1) The program randomly divides the dataset into ten folds; (2) [while looping over the tenfold each time:] save one fold as a ".arff" (i.e., WEKA supported file) file for later use (i.e., test set); (3) [while looping over the tenfold each time:] save the remaining folds (i.e., training set) as a ".arff" file; (4) After having saved all the test_set and training_set files, a bash script loop over the ten training set previously saved with a java command line that run an Auto-Weka analysis with it, and in addition perform a test over the best fitted model with a test set provided as parameter. Additionally, this entire process is then repeated for each dataset along the bash script, resulting in the tables shown below.

7.1.1 Pseudo-code of the Python Program refactored to save test/training set

Algorithm 4 Function that save the 10 random folds into training/test set for Auto-Weka usage.

```
1:  function CROSS_VALIDATION_PROCESS(all_10_folds, reSampling ← False)
2:
3:      for idx, fold ∈ enumerate(all_10_folds) do
4:          training_set ← all_10_folds.copy()
5:          training_set.pop(idx)
6:          test_set ← fold
7:
8:          input, output ← __get_inputs_outputs_from_folds(training_set)
9:          if reSampling then
10:             input, output ← reSamplingSMOTE(training_set, input, output)
11:         end if
12:         inputArffDf ← input.copy()
13:         output ← output.apply(str).astype("category")
14:         inputArffDf['class'] ← output
15:         trainingAttributes = self.save_set_arff(inputArffDf, "training_set_fold", str(idx), True)
16:         self.save_set_arff(test_set, "test_set_fold", str(idx), False, trainingAttributes)
17:     end for
18: end function
```

Algorithm 5 The Bash function that run auto-weka for each of the 10-fold (i.e, test_set and training_set) available following the python program execution.

```
1: function WEKACV
2:     for i ∈ 0..9 do
3:         FILE_TRAIN ← "./training_set_old_i.arff"
4:         FILE_TEST ← "./test_fold_i.arff"
5:         if FILE_TRAIN or FILE_TEST do not exist then
6:             continue
7:         end if
8:         java -cp autoweka.jar weka.classifiers.meta.AutoWEKAClassifier -t FILE_TRAIN -T
   FILE_TEST -timeLimit 60 -seed 85 -memLimit 5000 -nBestConfigs 50
9:     end for
10: end function
```

Table 11: Diabetic-retinopathy-Debrecen

Fold	HP	Classifier	Search Time (s)	Algorithm run (s)	Seed	Accuracy	Precision	recall	F1 score	AUROC
1	autoweka-diabetic-fold-1	AdaBoostM1	3008.29	0.01	85	66.6667	0.667	0.667	0.667	0.688
2	autoweka-diabetic-fold-2	Bagging	3050.73	0.04	85	62.2587	0.621	0.623	0.620	0.684
3	autoweka-diabetic-fold-3	Bagging	3031.84	0.05	85	59.1304	0.600	0.591	0.590	0.663
4	autoweka-diabetic-fold-4	Logistic	3009.51	0.03	85	64.6718	0.647	0.647	0.647	0.714
5	autoweka-diabetic-fold-5	LWL	3240.13	52.79	85	76.4479	0.770	0.764	0.765	0.857
6	autoweka-diabetic-fold-6	LWL	3220	49.8	85	89.5652	0.901	0.896	0.896	0.931
7	autoweka-diabetic-fold-7	Logistic	3008.71	0.1	85	74.7826	0.777	0.748	0.750	0.811
8	autoweka-diabetic-fold-8	Logistic	3009.45	0.03	85	74.7826	0.779	0.748	0.751	0.830
9	autoweka-diabetic-fold-9	LWL	3245.23	55.85	85	66.9565	0.674	0.670	0.670	0.757
10	autoweka-diabetic-fold-10	SMO	3273.63	0.13	85	77.7992	0.795	0.778	0.777	0.784
Average	N/A	N/A	3109.75	158.83	85.00	71.31	0.72	0.71	0.71	0.77

HP stands for Hyper parameter.

The metrics available here are the results of running the fitted model on unseen data using Auto-Weka.

Table 12: Obesity Dataset raw and data syntheti

Fold	HP	Classifier	Search Time (s)	Algorithm run (s)	Seed	Accuracy	Precision	recall	F1 score	AUROC
1	autoweka-obesity-fold-1	AdaBoostM1	3032.11	0.14	85	96.2264	0.964	0.962	0.962	0.999
2	autoweka-obesity-fold-2	AdaBoostM1	3026.3	0.13	85	96.2085	0.963	0.962	0.962	0.998
3	autoweka-obesity-fold-3	AdaBoostM1	3017.57	0.14	85	98.5782	0.987	0.986	0.986	0.999
4	autoweka-obesity-fold-4	Logistic	3059.73	0.04	85	78.673	0.787	0.787	0.786	0.967
5	autoweka-obesity-fold-5	AdaBoostM1	3069.91	0.12	85	97.6303	0.977	0.976	0.976	0.999
6	autoweka-obesity-fold-6	RandomForest	3122.27	0.06	85	91.4692	0.918	0.915	0.915	0.982
7	autoweka-obesity-fold-7	LMT	3162.25	0.04	85	86.7299	0.866	0.867	0.866	0.980
8	autoweka-obesity-fold-8	LMT	3047.7	0.03	85	97.6303	0.976	0.976	0.976	0.999
9	autoweka-obesity-fold-9	AdaBoostM1	3063.21	0.14	85	97.1223	0.974	0.971	0.971	1.000
10	autoweka-obesity-fold-10	Logistic	3095.53	0.06	85	80.5687	0.809	0.806	0.803	0.965
Average	N/A	N/A	3069.66	0.09	85.00	92.08	0.92	0.92	0.92	0.99

HP stands for Hyper parameter.

The metrics available here are the results of running the fitted model on unseen data using Auto-Weka.

Table 13: Thoracic Surgery Binary Survival

Fold	HP	Classifier	Search Time (s)	Algorithm run (s)	Seed	Accuracy	Precision	recall	F1 score	AUROC
1	autoweka-thoracic-fold-1	RandomTree	3007.24	0.03	85	85.1064	0.794	0.851	0.822	0.619
2	autoweka-thoracic-fold-2	RandomCommittee	3008.86	0.01	85	87.234	?	0.872	?	0.463
3	autoweka-thoracic-fold-3	SimpleLogistic	3007.98	0.02	85	80.8511	?	0.809	?	0.500
4	autoweka-thoracic-fold-4	KStar	3009.84	0.23	85	89.3617	?	0.894	?	0.674
5	autoweka-thoracic-fold-5	OneR	3007.39	0.02	85	91.4894	?	0.915	?	0.500
6	autoweka-thoracic-fold-6	Bagging	3012.99	0.01	85	87.234	?	0.872	?	0.492
7	autoweka-thoracic-fold-7	OneR	3006.57	0.01	85	100	1.000	1.00	1.000	?
8	autoweka-thoracic-fold-8	AdaBoostM1	3008.32	0.02	85	76.5957	0.616	0.766	0.683	0.584
9	autoweka-thoracic-fold-9	SimpleLogistic	3007.69	0.01	85	74.4681	?	0.745	?	0.500
10	autoweka-thoracic-fold-10	BayesNet	3007.64	0.01	85	80.8511	0.685	0.809	0.742	0.564
Average	N/A	N/A	3008.45	0.04	85.00	85.32	0.31	0.85	0.32	0.49

HP stands for Hyper parameter.

The metrics available here are the results of running the fitted model on unseen data using Auto-Weka.

Table 14: Breast cancer Coimbra

Fold	HP	Classifier	Search Time (s)	Algorithm run (s)	Seed	Accuracy	Precision	recall	F1 score	AUROC
1	autoweka-breast-cancer-fold-1	AdaBoostM1	3008.29	0.01	85	66.6667	0.667	0.667	0.667	0.688
2	autoweka-breast-cancer-fold-2	LMT	3077.06	0	85	50	0.500	0.500	0.500	0.391
3	autoweka-breast-cancer-fold-3	SMO	3007.96	0.02	85	75	0.844	0.750	0.745	0.786
4	autoweka-breast-cancer-fold-4	RandomForest	3008.07	0.01	85	83.3333	0.833	0.833	0.833	0.829
5	autoweka-breast-cancer-fold-5	AdaBoostM1	3007.8	0.01	85	75	0.764	0.750	0.752	0.886
6	autoweka-breast-cancer-fold-6	Bagging	3109.89	0.01	85	75	0.833	0.750	0.733	0.722
7	autoweka-breast-cancer-fold-7	Bagging	3406.07	0.01	85	72.7273	0.748	0.727	0.732	0.786
8	autoweka-breast-cancer-fold-8	Bagging	3008.06	0.01	85	72.7273	0.720	0.727	0.717	0.714
9	autoweka-breast-cancer-fold-9	Logistic	3134.68	0.01	85	63.6364	0.636	0.636	0.636	0.833
10	autoweka-breast-cancer-fold-10	RandomForest	3008.19	0.01	85	63.6364	0.636	0.636	0.636	0.194
Average	N/A	N/A	3077.61	0.01	85.00	69.77	0.72	0.70	0.70	0.68

HP stands for Hyper parameter.

The metrics available here are the results of running the fitted model on unseen data using Auto-Weka.

Table 15: Heart failure clinical records

Fold	HP	Classifier	Search Time (s)	Algorithm run (s)	Seed	Accuracy	Precision	recall	F1 score	AUROC
1	autoweka-heart-failure-fold-1	MultilayerPerceptron	3010.13	0.01	85	83.3333	0.839	0.833	0.835	0.860
2	autoweka-heart-failure-fold-2	RandomForest	3005.26	0.02	85	97.7695	0.978	0.978	0.977	0.999
3	autoweka-heart-failure-fold-3	JRip	3002.75	0.01	85	76.6667	0.771	0.767	0.768	0.817
4	autoweka-heart-failure-fold-4	RandomCommittee	3000.49	0.1	85	83.3333	0.841	0.833	0.836	0.904
5	autoweka-heart-failure-fold-5	OneR	3004.89	0.01	85	80	0.814	0.800	0.790	0.764
6	autoweka-heart-failure-fold-6	BayesNet	3002.79	0.01	85	83.3333	0.834	0.833	0.833	0.882
7	autoweka-heart-failure-fold-7	SMO	3002.36	0.05	85	73.3333	0.707	0.733	0.717	0.578
8	autoweka-heart-failure-fold-8	J48	3003.28	0.02	85	86.6667	0.867	0.867	0.867	0.696
9	autoweka-heart-failure-fold-9	RandomSubSpace	3000.69	0.03	85	93.3333	0.933	0.933	0.933	0.942
10	autoweka-heart-failure-fold-10	RandomTree	3010.06	0.02	85	82.7586	0.834	0.828	0.821	0.785
Average	N/A	N/A	3004.27	0.03	85.00	84.05	0.84	0.84	0.84	0.82

HP stands for Hyper parameter.

The metrics available here are the results of running the fitted model on unseen data using Auto-Weka.

References

[1] M. Zikeba, J. M. Tomczak, M. Lubicz, and J. 'Swikatek, "Boosted svm for extracting rules from imbalanced data in application to prediction of the post-operative life expectancy in the lung cancer patients," *Applied Soft Computing*, 2013.

[2] B. Antal and A. Hajdu, "An ensemble-based system for automatic screening of diabetic retinopathy," *Knowledge-based systems*, vol. 60, pp. 20–27, 2014.

[3] F. M. Palechor and A. de la Hoz Manotas, "Dataset for estimation of obesity levels based on eating habits and physical condition in individuals from colombia, peru and mexico," *Data in brief*, vol. 25, p. 104344, 2019.

[4] M. Patrício, J. Pereira, J. Crisóstomo, P. Matafome, M. Gomes, R. Seiça, and F. Caramelo, "Using resistin, glucose, age and bmi to predict the presence of breast cancer," *BMC cancer*, vol. 18, no. 1, pp. 1–8, 2018.

[5] D. Chicco and G. Jurman, "Machine learning can predict survival of patients with heart failure from serum creatinine and ejection fraction alone," *BMC medical informatics and decision making*, vol. 20, no. 1, pp. 1–16, 2020.

[6] M. Feurer, A. Klein, K. Eggensperger, J. Springenberg, M. Blum, and F. Hutter, "Efficient and robust automated machine learning, 2015," *URL http://papers. nips. cc/paper/5872-efficient-and-robust-automated-machine-learning*.