Cherry picking evaluation metrics

The goal of this task is to learn how performance assessments may vary based on the evaluation metric. You shall also learn that existing performance assessments of machine-learning algorithms must be looked at with some skepticism. For this task, we will assume a scenario in which you developed two novel algorithms and you want to show that they perform better than the state-of-the-art.

More precisely, your goal is to answer the following question:

1. How meaningful are evaluation metrics ("meaningful" in terms of how consistent are they in assessing the performance of machine-learning algorithms)? [200-300 words]

Details:

1. Choose four algorithms of your choice for either a regression/prediction *or* a classification task (feel free to use multiple frameworks if necessary).
    a. Out of the four algorithms, randomly pick one algorithm that we will consider as "novel" (you can use whatever existing algorithm you want even if it's implemented in e.g. scikit-learn since 10 years – we just assume it is novel for the sake of the assignment). Please, really pick this algorithm at random, and stick to it whatever results the evaluation shows!
    b. The remaining three algorithms are our "baselines" to which we compare the "novel" algorithm.
2. Choose two datasets, each with at least 10,000 data points. Each dataset must be suitable for the novel algorithm and at least two of the three baseline algorithms.
3. Use 10-fold cross validation (use another evaluation method if that gives any benefits to you).
4. Calculate five common evaluation metrics of your choice. Feel free to calculate more, but only include the five most "diverse" in your results. By "diverse" we mean e.g. one metric that shows that your novel algorithm is performing very well, one showing that your algorithm is performing mediocre, and another metric showing that your algorithm is performing poorly.
5. Make up two new evaluation metrics. Both metrics should be "reasonable", meaning that someone else would think "ok, that metric might makes sense". One metric should show that your novel algorithm performs (far) better than the baselines, and one metric shall show that your novel algorithm performs (far) worse than the baselines. For instance, calculate metrics for different user groups and weight the results for the groups in a "favorable" way. This way, you would be able to argue "Our novel algorithm might not be the most effective algorithm overall, but it performed really well for young males from Germany, which is our primary target group". Or, create some "runtime-effectiveness" metric that expresses how effective the algorithm is per second CPU time, and hence would allow you drawing a conclusion like "Our novel algorithm might not be the most effective algorithm overall. However, the value for money is best as it has the highest e.g. accuracy per CPU second". Please note that it is fine, if you can't find any metrics that make your novel algorithms to be best or worst performing. If you calculate five standard evaluation metrics, and try unsuccessfully to make up some metrics, this will not affect your marks. Actually, it would also be interesting to learn if some algorithms were very robust to changes in the evaluation metrics.

6. Count how often which of the four algorithms algorithm performed best, second best, third best, … over all metrics and datasets.

Deliverables:

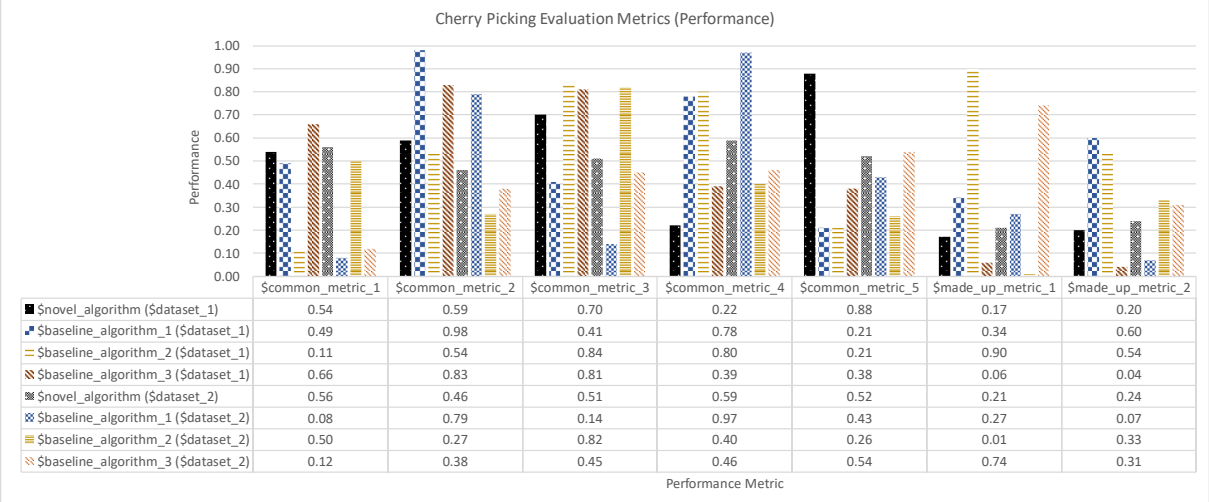
1. A CSV file with your results. The CSV file must be exactly (!) in the format as specified by the example file. The file name must follow exactly (!) the pattern “task3, $team_id, cherry evaluation, data.csv”, e.g. “task3, team_01, cherry evaluation, data.csv”. The first part of the table shows how the algorithms performed as measured with the seven metrics. The second part shows how an algorithm ranked (best, second best, third best, …) when compared with a particular metric to the other algorithms. The third part shows how often overall the algorithms were ranked best, second best, …

Performance	$novel_algorithm ($dataset_1)	$baseline_algorithm_1 ($dataset_1)	$baseline_algorithm_2 ($dataset_1)	$baseline_algorithm_3 ($dataset_1)	$novel_algorithm ($dataset_2)	$baseline_algorithm_1 ($dataset_2)	$baseline_algorithm_2 ($dataset_2)	$baseline_algorithm_3 ($dataset_2)
$common_metric_1	0.67	0.25	0.84	0.71	0.09	0.73	0.73	0.95
$common_metric_2	0.8	0.05	0.73	0.91	0.32	0.04	0.06	0.7
$common_metric_3	0.2	0.19	0.9	1	0.73	0.27	0.68	0.38
$common_metric_4	0.93	0.7	0.76	0.13	0.87	0.48	0.48	0.84
$common_metric_5	1	0.36	0.43	0.74	0.73	0.4	0.31	0.78
$made_up_metric_1	0.31	0.05	0.15	0.25	0.87	0.69	1	0.18
$made_up_metric_2	0.16	0.97	0.93	0.1	0.74	0.84	0.23	0.86

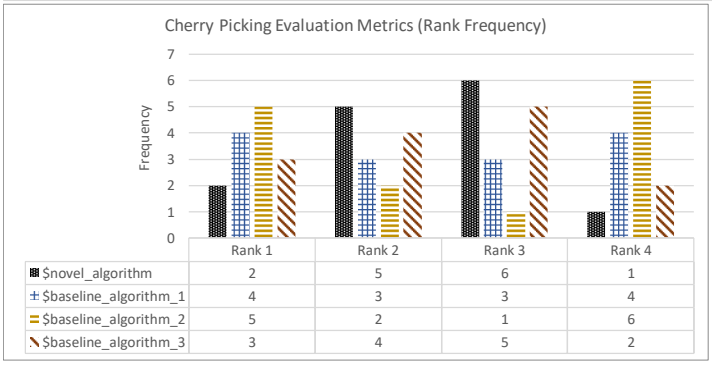
Rank	$novel_algorithm ($dataset_1)	$baseline_algorithm_1 ($dataset_1)	$baseline_algorithm_2 ($dataset_1)	$baseline_algorithm_3 ($dataset_1)	$novel_algorithm ($dataset_2)	$baseline_algorithm_1 ($dataset_2)	$baseline_algorithm_2 ($dataset_2)	$baseline_algorithm_3 ($dataset_2)
$common_metric_1	3	4	1	2	4	2	2	1
$common_metric_2	2	4	3	1	2	4	3	1
$common_metric_3	3	4	2	1	1	4	2	3
$common_metric_4	1	3	2	4	1	3	3	2
$common_metric_5	1	4	3	2	2	3	4	1
$made_up_metric_1	1	4	3	2	2	3	1	4
$made_up_metric_2	3	1	2	4	3	2	4	1

Rank Frequency (Overall)	$novel_algorithm	$baseline_algorithm_1	$baseline_algorithm_2	$baseline_algorithm_3
Rank 1	5	1	2	6
Rank 2	4	2	5	4
Rank 3	4	4	5	1
Rank 4	1	7	2	3

2. Two PNG images illustrating your results, named “task3, $team_id, cherry evaluation, chart (performance).png” and “task3, $team_id, cherry evaluation, chart (rank frequency).png”. The two images should look like this:



	$common_metric_1	$common_metric_2	$common_metric_3	$common_metric_4	$common_metric_5	$made_up_metric_1	$made_up_metric_2
$novel_algorithm ($dataset_1)	0.54	0.59	0.70	0.22	0.88	0.17	0.20
$baseline_algorithm_1 ($dataset_1)	0.49	0.98	0.41	0.78	0.21	0.34	0.60
$baseline_algorithm_2 ($dataset_1)	0.11	0.54	0.84	0.80	0.21	0.90	0.54
$baseline_algorithm_3 ($dataset_1)	0.66	0.83	0.81	0.39	0.38	0.06	0.04
$novel_algorithm ($dataset_2)	0.56	0.46	0.51	0.59	0.52	0.21	0.24
$baseline_algorithm_1 ($dataset_2)	0.08	0.79	0.14	0.97	0.43	0.27	0.07
$baseline_algorithm_2 ($dataset_2)	0.50	0.27	0.82	0.40	0.26	0.01	0.33
$baseline_algorithm_3 ($dataset_2)	0.12	0.38	0.45	0.46	0.54	0.74	0.31



	Rank 1	Rank 2	Rank 3	Rank 4
$novel_algorithm	2	5	6	1
$baseline_algorithm_1	4	3	3	4
$baseline_algorithm_2	5	2	1	6
$baseline_algorithm_3	3	4	5	2

3. A PDF report that
    a. Contains the team id, individual student IDs, and the total time that was spent to complete the task
    b. Answers the above-mentioned questions

   c. Contains the created image (please ensure that the quality of the image is good. It doesn't matter if we must zoom in e.g. 400%, but the numbers etc. must be readable at some zoom level).
   d. Provides a brief overview (table) of the used datasets, algorithms, etc.
   e. Lists the contributions of all team members
   f. Contains additional information if required
   g. Is in about the same format as the report-template and named "task2, $team_id, consistency of ml frameworks, report.pdf"
4. Your source code (no compiled/binary files) in the sub folder / task2, $team_id, consistency of ml frameworks, source/. Ensure you follow the "developer guidelines".