Title: Pairwise Difference Learning

Speakers: Karim Belaid

Collection/Series: Machine Learning Initiative

Subject: Other

Date: November 29, 2024 - 2:30 PM

URL: https://pirsa.org/24110070

Abstract:

Pairwise difference learning (PDL) has recently been introduced as a new meta-learning technique for regression by Wetzel et al. Instead of learning a mapping from instances to outcomes in the standard way, the key idea is to learn a function that takes two instances as input and predicts the difference between the respective outcomes. Given a function of this kind, predictions for a query instance are derived from every training example and then averaged. This presentation focus on the classification version of PDL, proposing a meta-learning technique for inducing a classifier by solving a suitably defined (binary) classification problem on a paired version of the original training data. This presentation will also discuss an enhancement to PDL through anchor weighting, which adjusts the influence of anchor points based on the reliability and precision of their predictions, thus improving the robustness and accuracy of the method. We analyze the performance of the PDL classifier in a large-scale empirical study, finding that it outperforms state-of-the-art methods in terms of prediction performance. Finally, we provide an easy-to-use and publicly available implementation of PDL in a Python package.

Pairwise Difference Learning

Karim Belaid[1,2,3], Tim Wibiral[2,4], Maximilian Rabus[2], Eyke Hüllermeier[3]

[1] IDIADA GmbH [2] Porsche AG [3] LMU [4] Ulm University

2024-11-29

--- In Brief ---

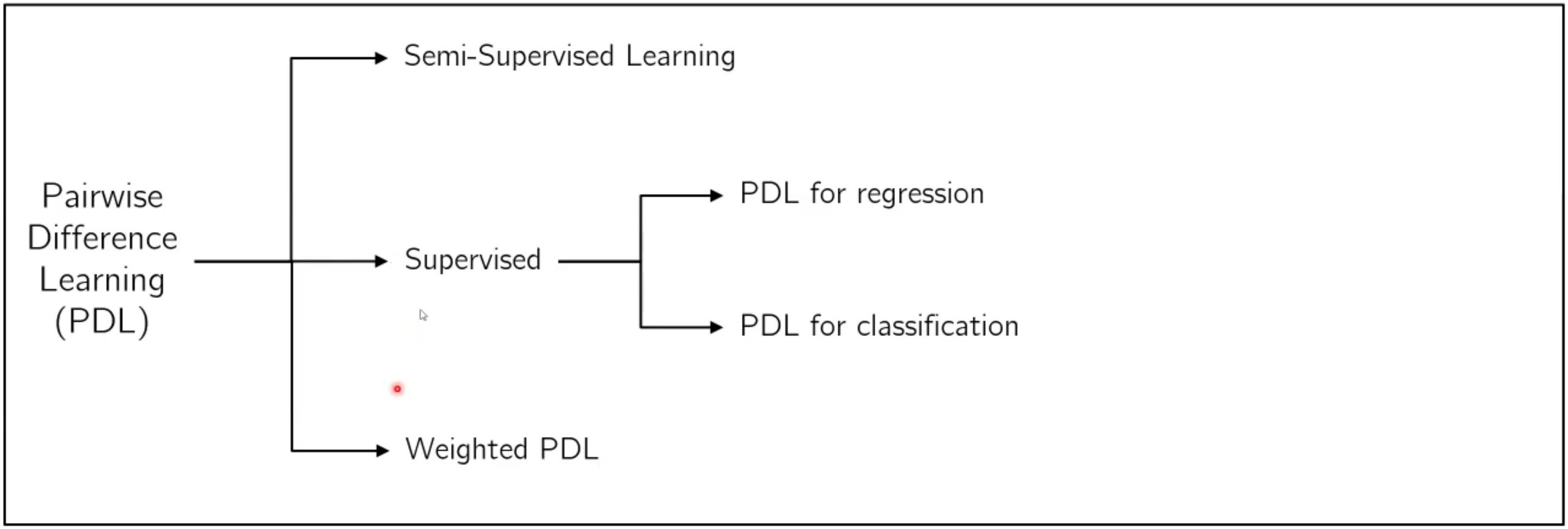
Pairwise Difference Learning (PDL)

- is a new sklearn-compatible ML algorithm
- for **regression** and **classification** tasks (for now).
- PDL outperforms state-of-the-art ML algorithms tested on tabular datasets < 5 000 datapoints
- Available Scikit-learn compatible library:

```
!pip install pdll
```

— OVERVIEW —

Pairwise Difference Learning (PDL) is a ML meta-algorithm that learns the difference between pairs of inputs, rather than their absolute values, leading to better performance.



— Our contribution: PDL for Classification —

- 1. Learn similarity g_{sym}: $\mathbb{R}^{2d} \rightarrow \mathbb{R}$

$$(x_i, x_j) \mapsto y_{i,j} = \begin{cases} 0 & \text{for } y_i \neq y_j, \\ 1 & \text{for } y_i = y_j \end{cases}$$

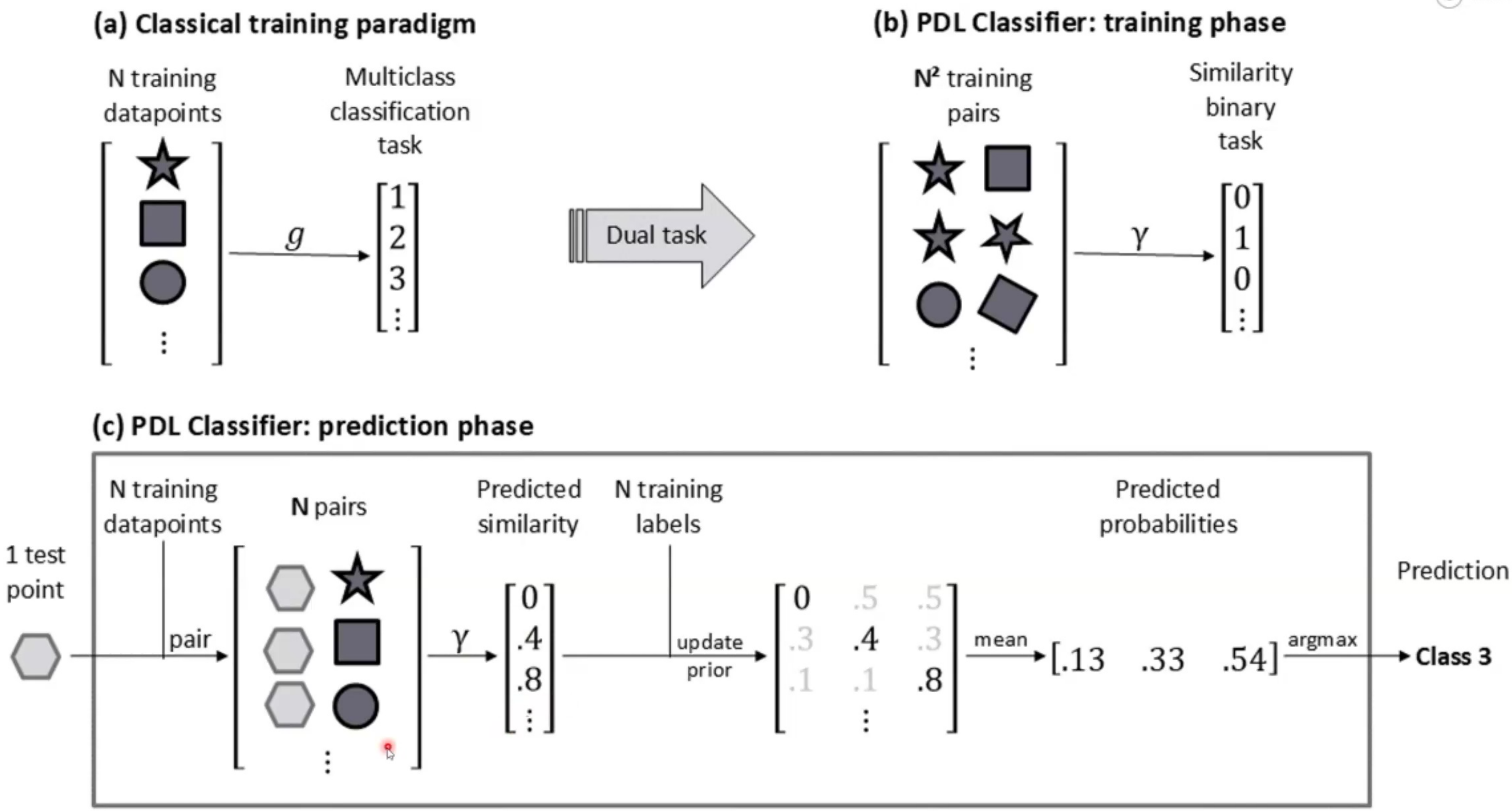
- 2. Query point:

$$p_{post,i}(y) = \begin{cases} g_{sym}(x_q, x_i) & \text{if } y = y_i \\ \dfrac{p(y) \cdot (1 - g_{sym}(x_q, x_i))}{1 - p(y_i)} & \text{otherwise} \end{cases}$$

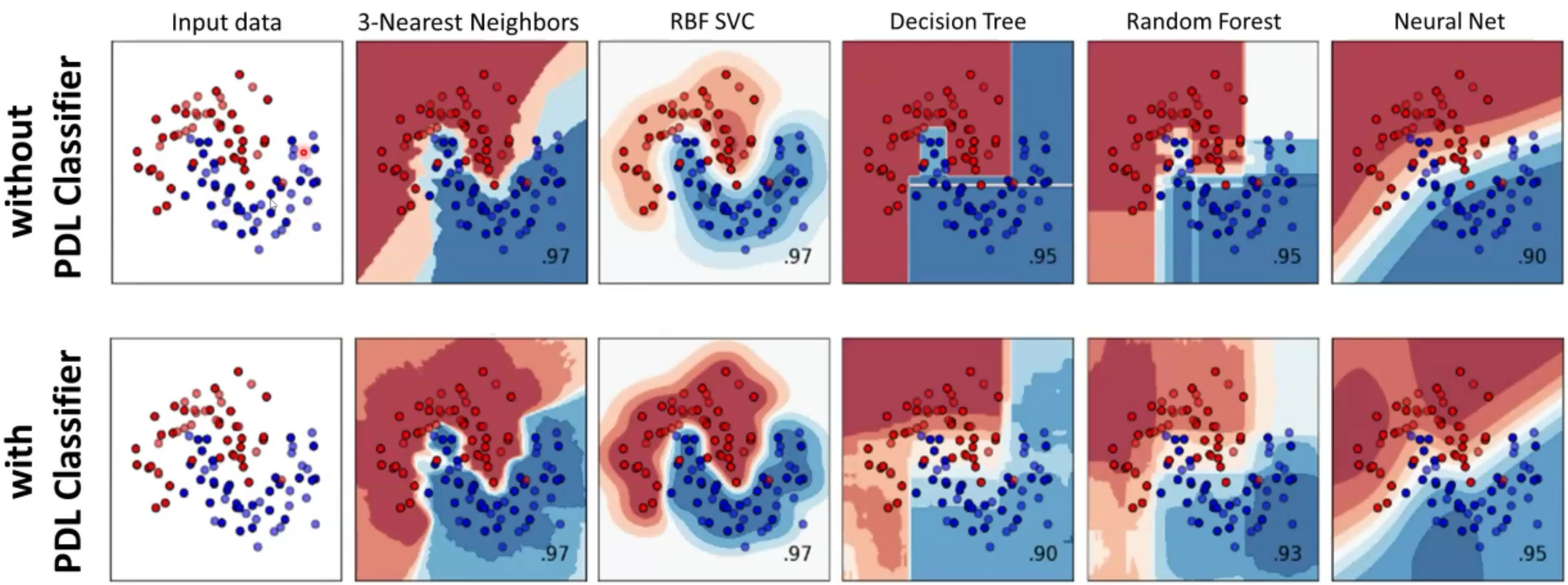
$$p_{post}(y) = \frac{1}{N} \sum_{i=1}^{N} p_{post,i}(y) \, .$$

anchor points

— OUR CONTRIBUTION: PDL FOR CLASSIFICATION —



Boundary Analysis



--- EVALUATION ---

- Benchmark: 99 OpenML datasets, 5 times 5-fold cross-validation.
- Parameter Optimization: Grid Search CV 3-folds
- Baselines: 7 state-of-the-art ML models (Random Forest, Extra Trees, Gradient Boosting, Bagging, etc.)

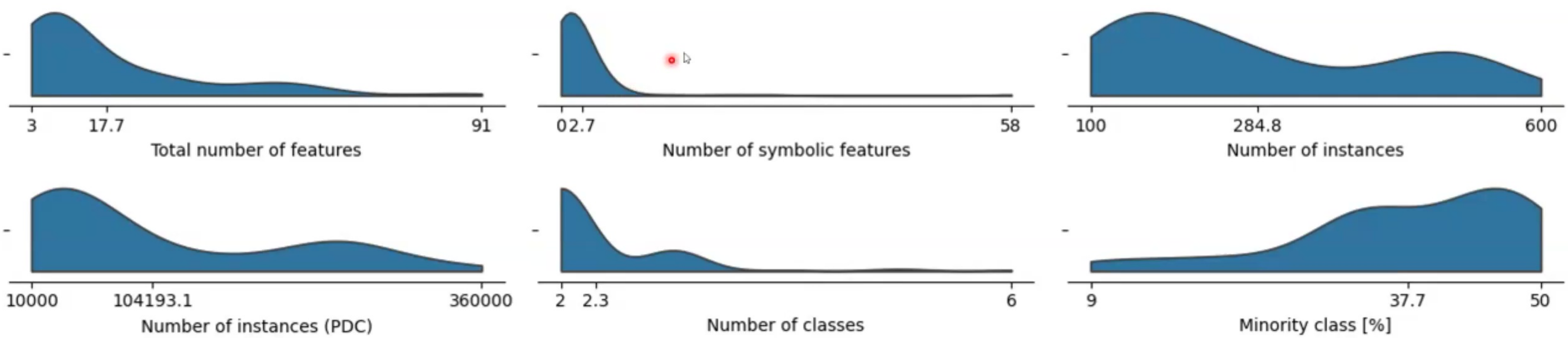
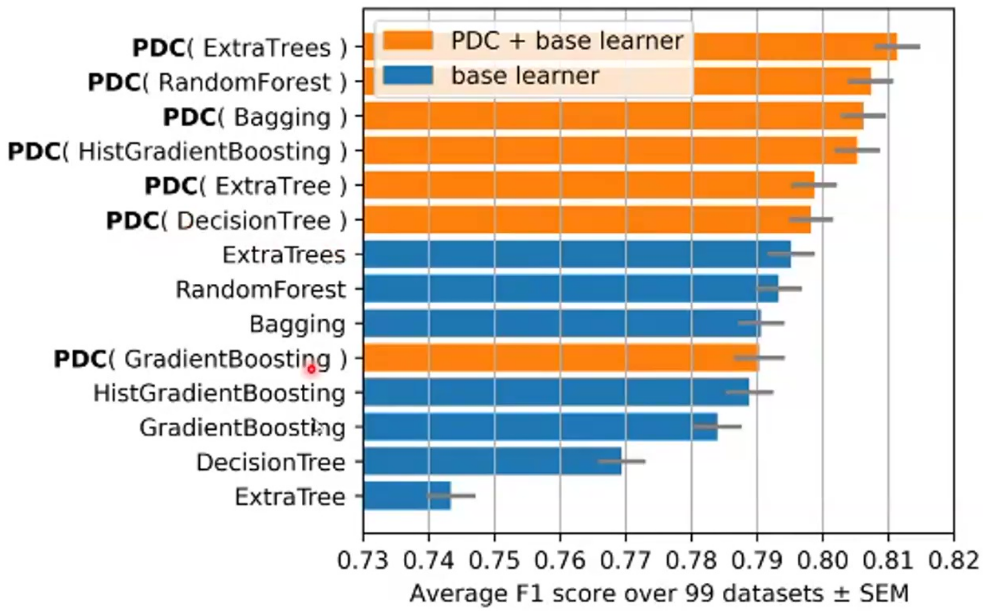


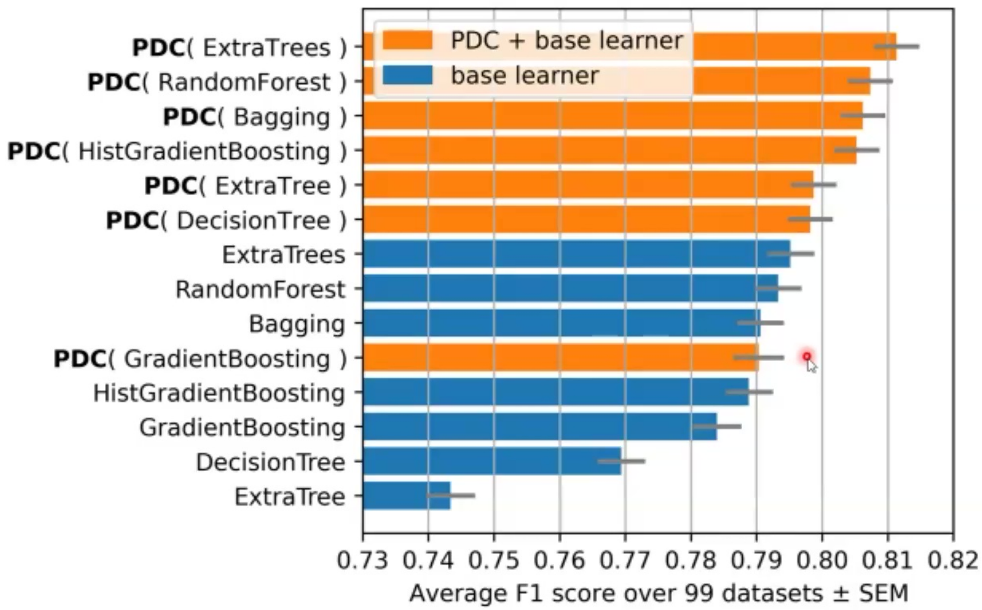
Figure: Distribution of key characteristics of the 99 OpenML classification datasets used for evaluation, specifying the minimum, mean, and maximum.

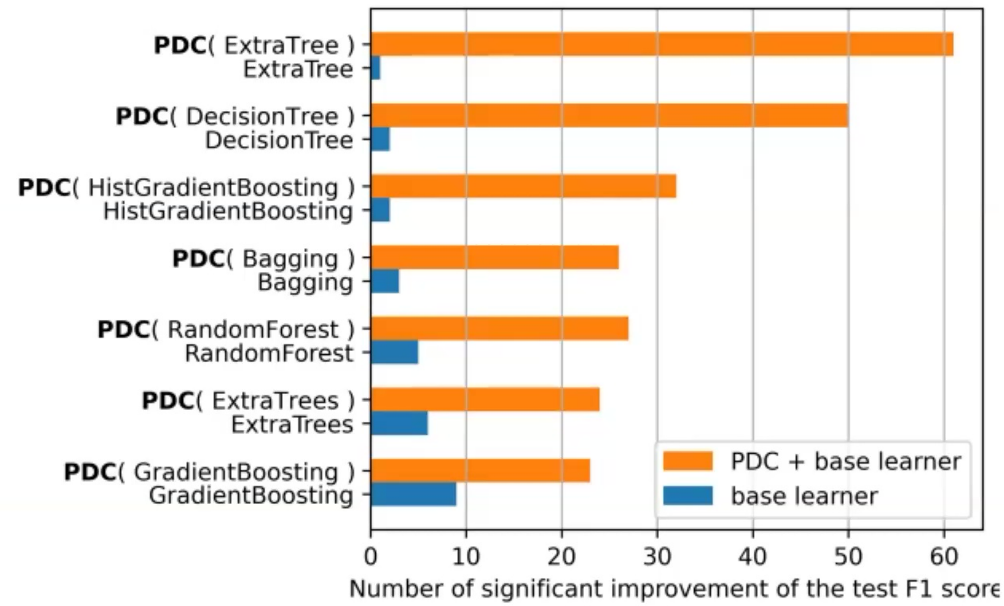
--- RESULTS ---



- PDC(ExtraTrees) obtains the best results.
- ExtraTrees is better than RandomForest.
- GradientBoosting is not a good base learner.

— RESULTS —


PDC + base learner
base learner
PDC(ExtraTrees)
PDC(RandomForest)
PDC(Bagging)
PDC(HistGradientBoosting)
PDC(ExtraTree)
PDC(DecisionTree)
ExtraTrees
RandomForest
Bagging
PDC(GradientBoosting)
HistGradientBoosting
GradientBoosting
DecisionTree
ExtraTree
0.73 0.74 0.75 0.76 0.77 0.78 0.79 0.80 0.81 0.82
Average F1 score over 99 datasets ± SEM


PDC(ExtraTree)
ExtraTree
PDC(DecisionTree)
DecisionTree
PDC(HistGradientBoosting)
HistGradientBoosting
PDC(Bagging)
Bagging
PDC(RandomForest)
RandomForest
PDC(ExtraTrees)
ExtraTrees
PDC(GradientBoosting)
GradientBoosting
PDC + base learner
base learner
0 10 20 30 40 50 60
Number of significant improvement of the test F1 score

— WHY DOES PDL YIELD IMPROVED PERFORMANCE? —

1. **Model-based Learning**:
   Learning the difference might lead to easier pattens.

I know the price of this house from the training set

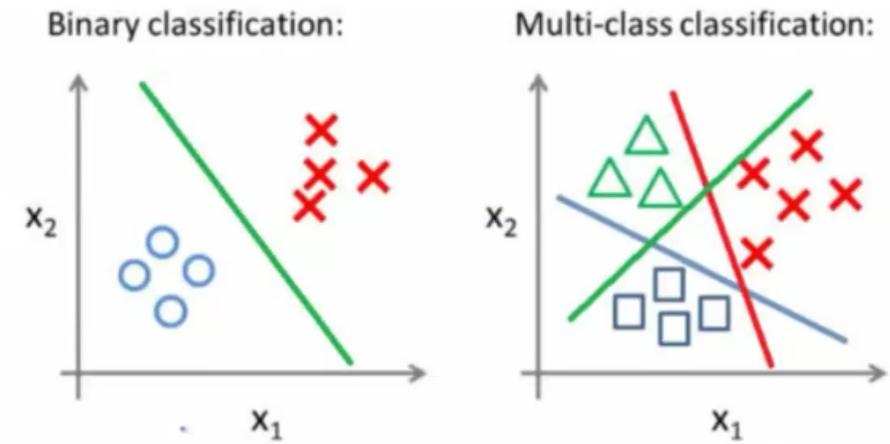
This bigger house must be slightly more expansive

Why Does PDL Yield Improved Performance?

1. **Model-based Learning**:
   Learning the difference might lead to easier pattens.

2. **Combining Instance-based**:
   a prediction for a new query is produced by an ensemble of instances from the training set.

3. **Simplification through Binary Reduction**:
   Available training instances contribute to building one model.



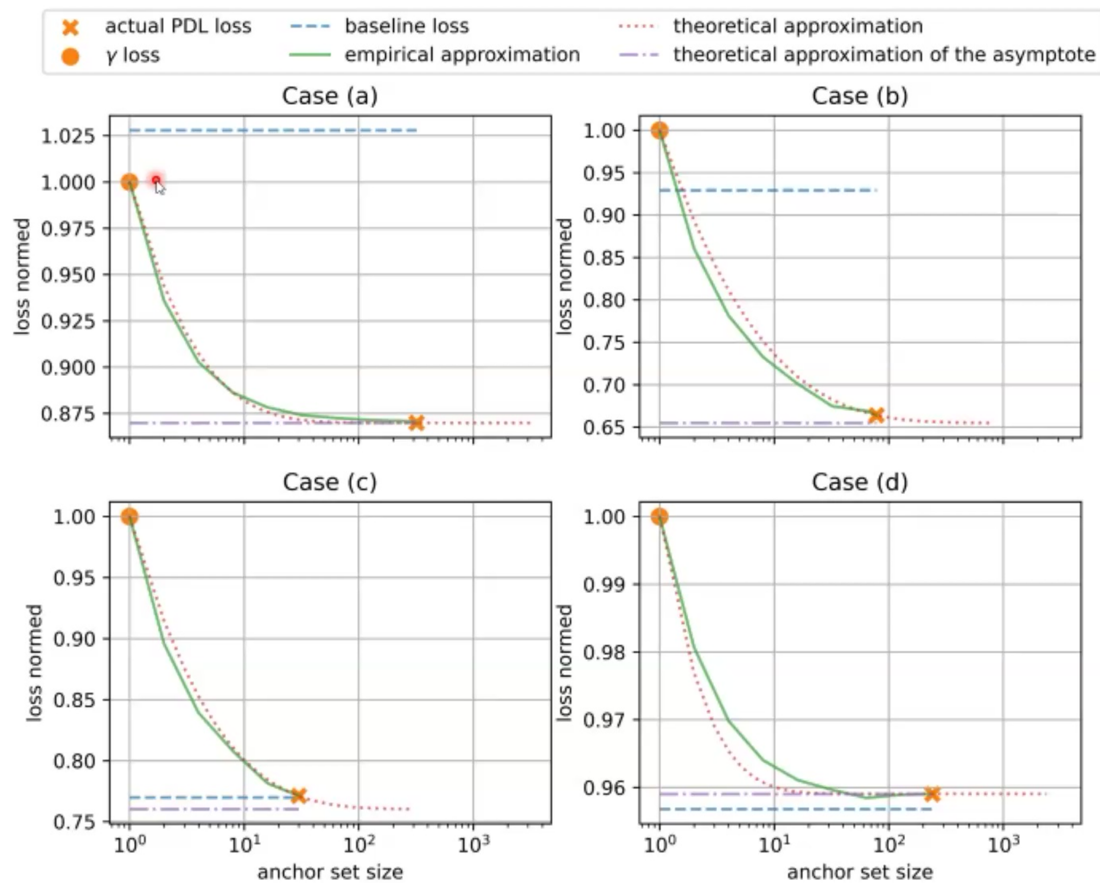
source: utkuufuk.com

— Why Does PDL Yield Improved Performance? —

1. **Model-based Learning**: Learning the difference might lead to easier pattens.

2. **Combining Instance-based**: a prediction for a new query is produced by an ensemble of instances from the training set.

3. **Simplification through Binary Reduction**: Available training instances contribute to building one model.



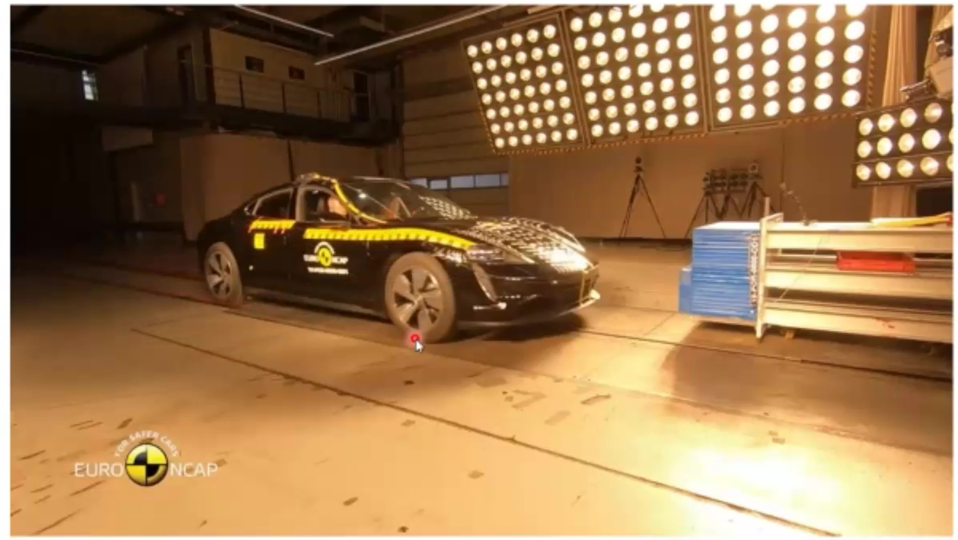
- Motivation to use PDL:

Real crash tests are expansive ~ 500 000 €

Electric cars represent a data shift in ML

- Results:

Improvement of the ML models used at Porsche to predict the safety of car prototypes



Euro NCAP Crash Test of Porsche Taycan 2019

source: youtu.be/wEzPaH1xhPA?si=IdcyNuwJq2pCWG72

Find more about this project in this paper:
Rabus, M., Belaid, M. K., Maurer, S. A., & Hiermaier, S. (2022). Development of a model for the prediction of occupant loads in vehicle crashes: introduction of the Real Occupant Load Criterion for Prediction ($ROLC_p$). Automotive and Engine Technology, 7(3), 229-244.

— PDL Library —

```
!pip install pdll
from pdll import PairwiseDifferenceClassifier
X, y = load_data()
model = RandomForestClassifier()
model = PairwiseDifferenceClassifier(model)
model.fit(X, y)
```

— WHEN TO USE PDL ? —

- ✓ You are working on one of the following problems:
  Classification, **Regression**, Semi supervised learning,

- ✓ You have a relatively small training set:
  between 10 and 10 000 training points

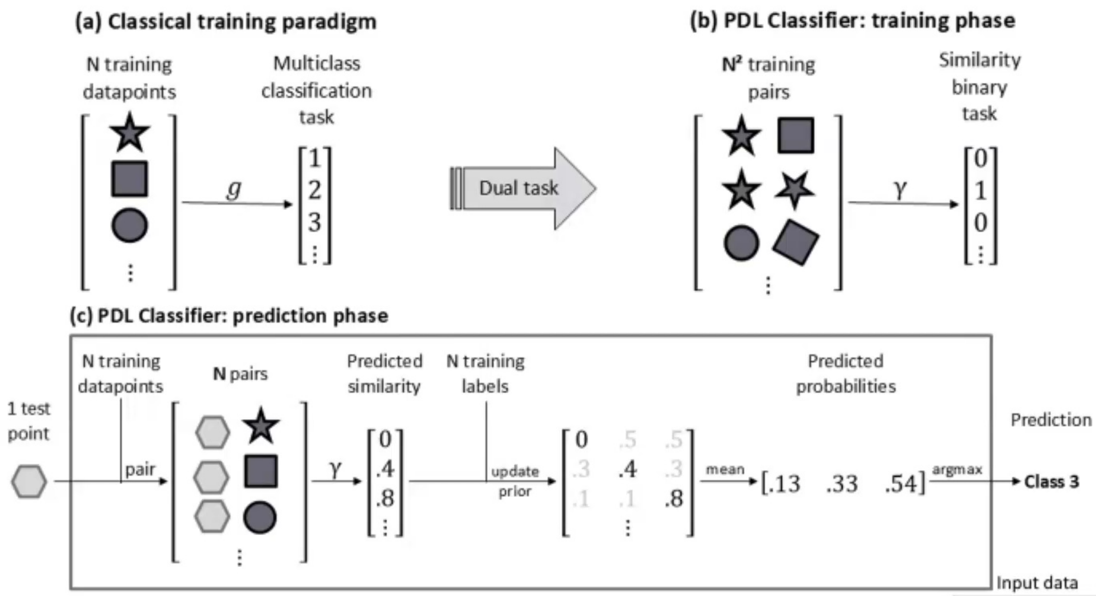
- ✓ The input features are:
  Tabular data, text, graphs, images

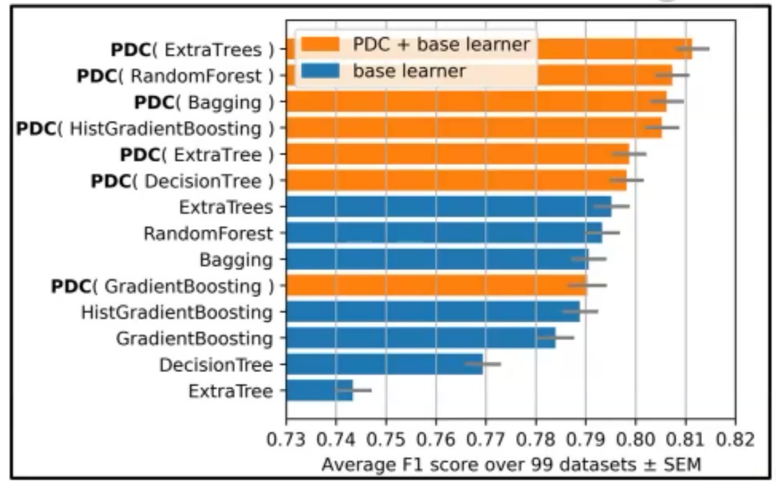
- ✓ You need to improve the performance of the current model:
  There is still room to improve the performance

— Related Work —

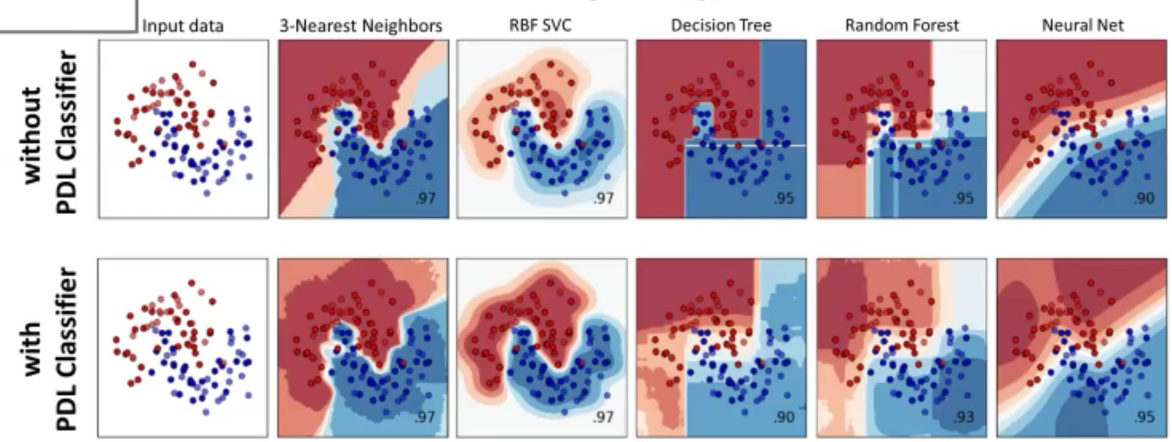
	Task			Input features			models		Anchors
	regression	classification	semi-supervised	x1,x2	x1-x2	|x1-x2|	NN	Scikit learn	Weighting method
Wetzel et al.	✓		✓	✓	✓		✓	✓	k-NN
Tynes et al.	✓			✓	✓			✓	Average
Corbara et al.		✓				✓	✓		Model
Siamese Net	✓	✓		✓			✓		
Belaid et al.		✓		✓	✓			✓	Optimizer

— Q&A —





Boundary Analysis



— Related Work —

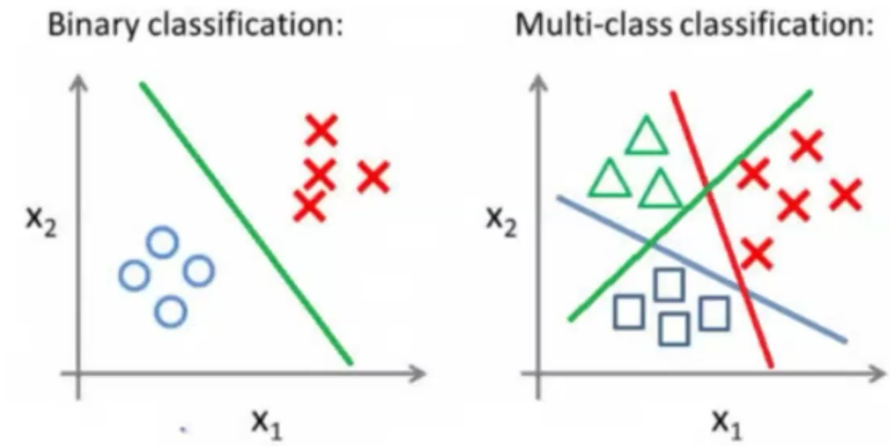
	Task			Input features			models		Anchors
	regression	classification	semi-supervised	x1,x2	x1-x2	|x1-x2|	NN	Scikit learn	Weighting method
Wetzel et al.	✓		✓	✓	✓		✓	✓	k-NN
Tynes et al.	✓			✓	✓			✓	Average
Corbara et al.		✓				✓	✓		Model
Siamese Net	✓	✓		✓			✓		
Belaid et al.		✓		✓	✓			✓	Optimizer

— WHY DOES PDL YIELD IMPROVED PERFORMANCE? —

1. **Model-based Learning**:
   Learning the difference might lead to easier pattens.

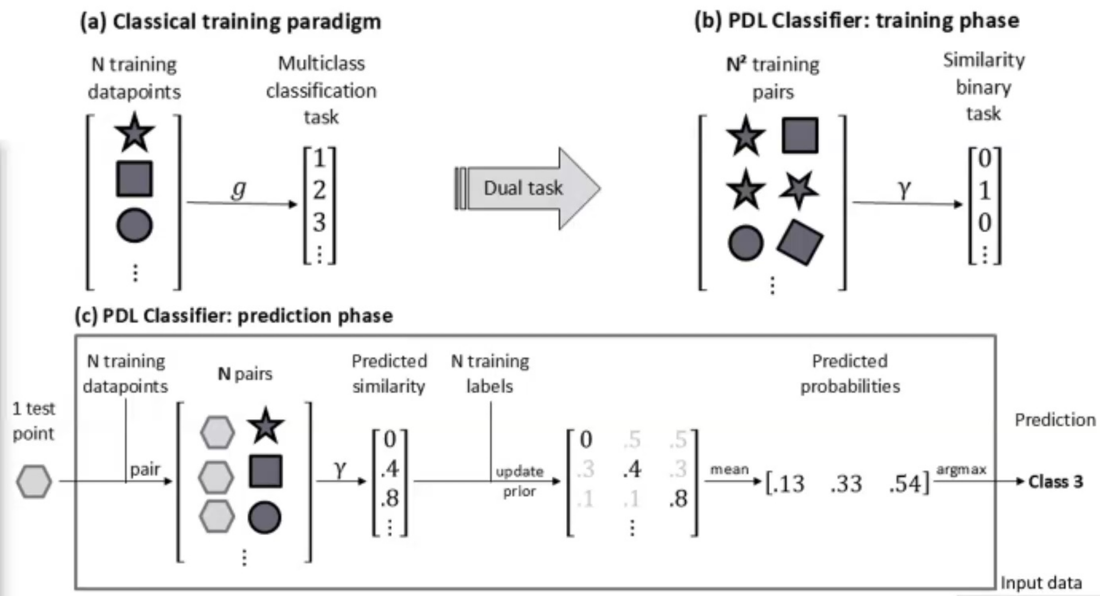
2. **Combining Instance-based**:
   a prediction for a new query is produced by an ensemble of instances from the training set.

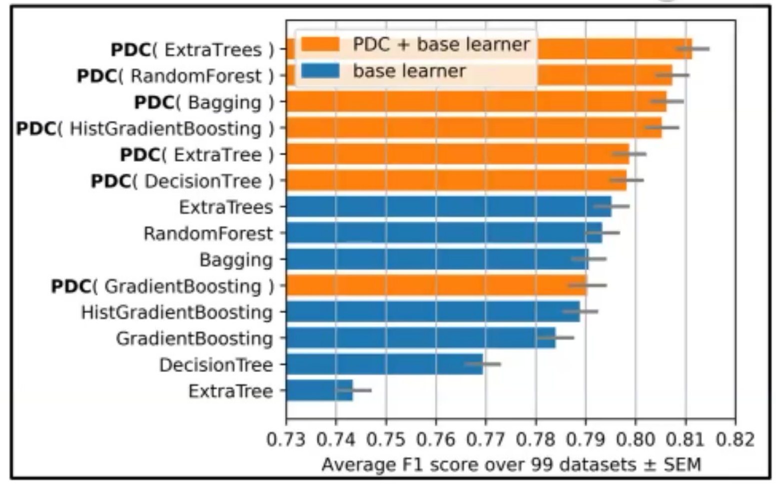
3. **Simplification through Binary Reduction**:
   Available training instances contribute to building one model.



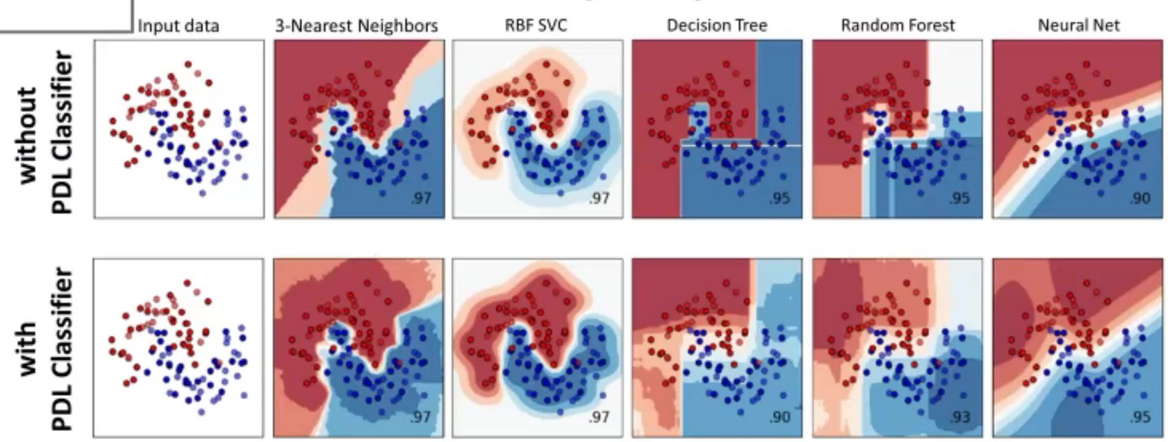
source: utkuufuk.com

— Q&A —





Boundary Analysis

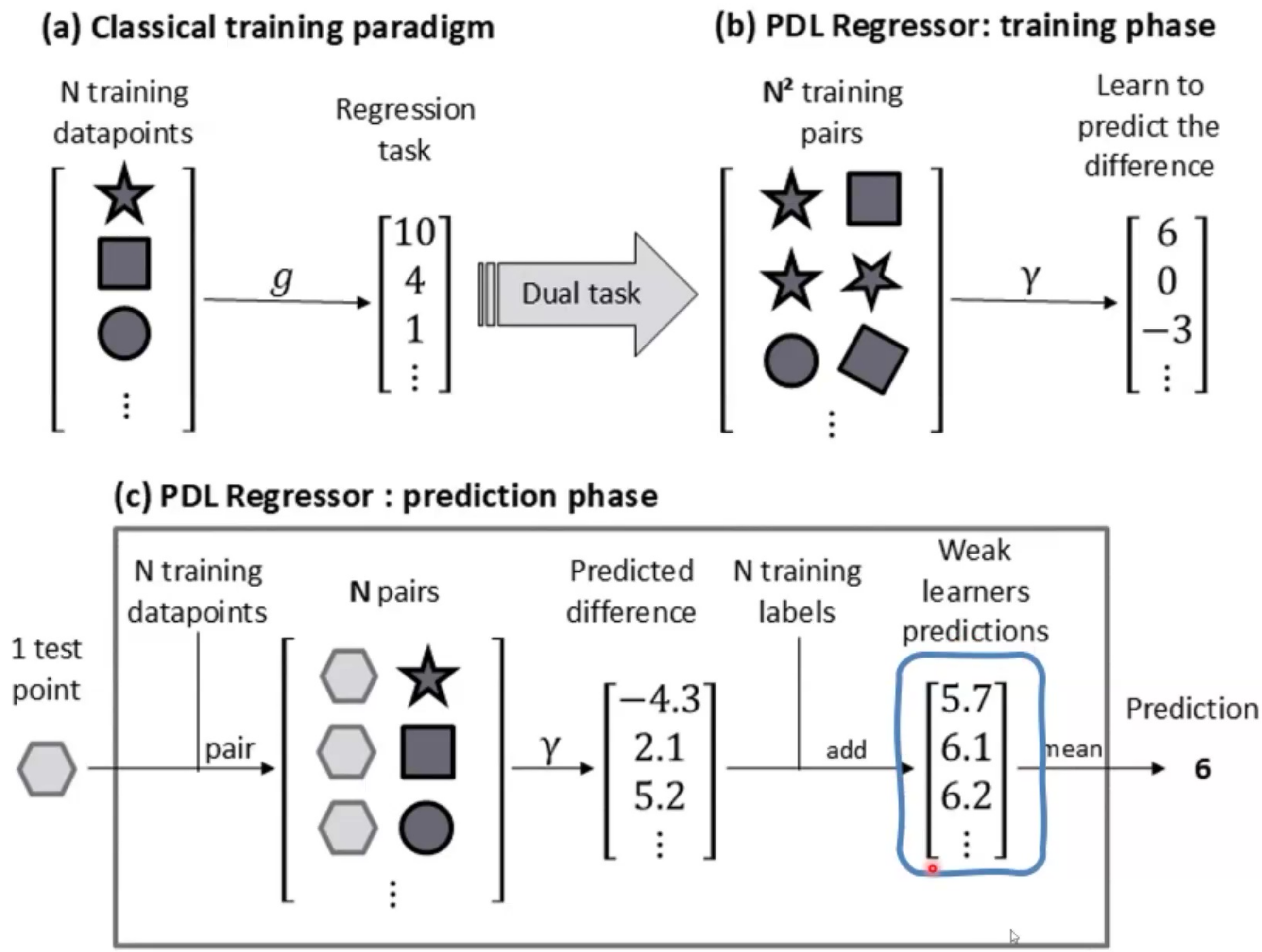


PDL LIBRARY

```
!pip install pdll
from pdll import PairwiseDifferenceClassifier
X, y = load_data()
model = RandomForestClassifier()
model = PairwiseDifferenceClassifier(model)
model.fit(X, y)
```

Star

— WEIGHTED PDL —



--- Method: Weighted PDL ---

- Objective function: Minimize the validation loss $$L(w) = \frac{1}{|\mathscr{D}_{val}|} \sum_{(x_q, y_q) \in \mathscr{D}_{val}} \left| y_q - \hat{y}_q(w) \right| ,$$

- and the regularization terms: L_1- norm L_2- norm Kullback–Leibler divergence (KLD)

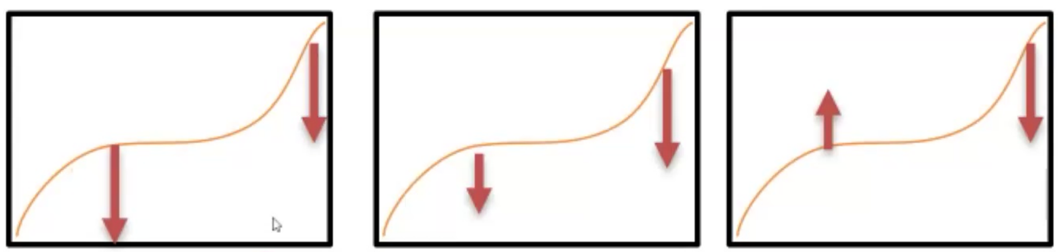
$$w^{\text{opt}} = \arg \min_{w \in [0,1]^k} L(w) \underbrace{- \lambda_1 \cdot L(u) \cdot w_{\max}}_{L_1} \underbrace{+ \lambda_2 \cdot L(u) \cdot \|w\|_2^2}_{L_2} \underbrace{+ \lambda \cdot L(u) \cdot \text{KL}(w \parallel u)}_{\text{KLD}} \quad \textbf{s.t.} \quad \|w\|_1 = 1 \,.$$

Initial loss with a uniform distribution

--- METHOD: WEIGHTED PDL ---

- Objective function: Minimize the validation loss $L(w) = \frac{1}{|\mathscr{D}_{val}|} \sum_{(x_q, y_q) \in \mathscr{D}_{val}} \left| y_q - \hat{y}_q(w) \right|$,

- and the regularization terms: L_1- norm L_2- norm Kullback–Leibler divergence (KLD)

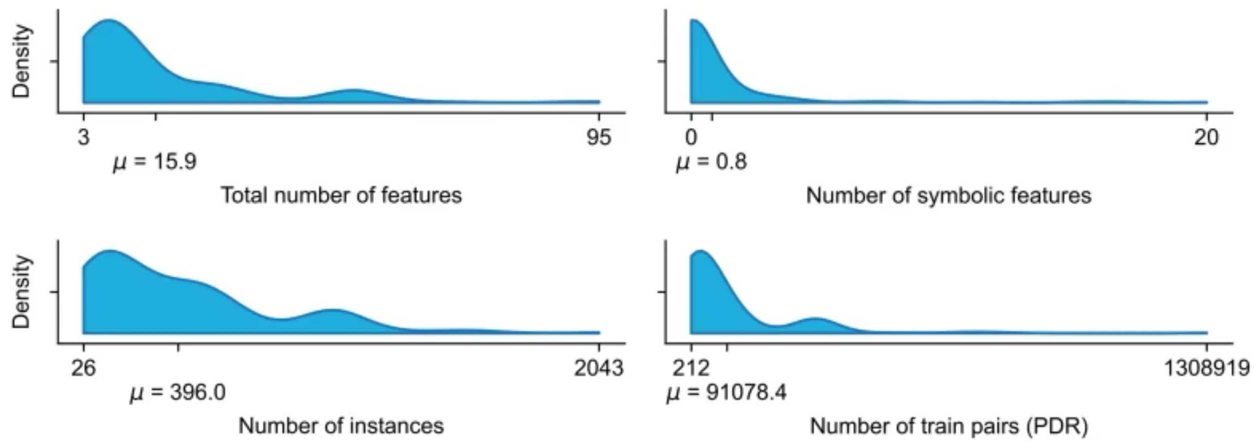


--- Evaluation ---

- Train PDL on the training set
  Learn the weights on the validation set
  Optimal weights are assessed on the testing set

- Benchmark: 231 OpenML datasets, 5 times 5-fold cross-validation.

- Baselines: 7 state-of-the-art ML models (Random Forest, Extra Trees, Gradient Boosting, Bagging, etc.)



--- RESULTS ---

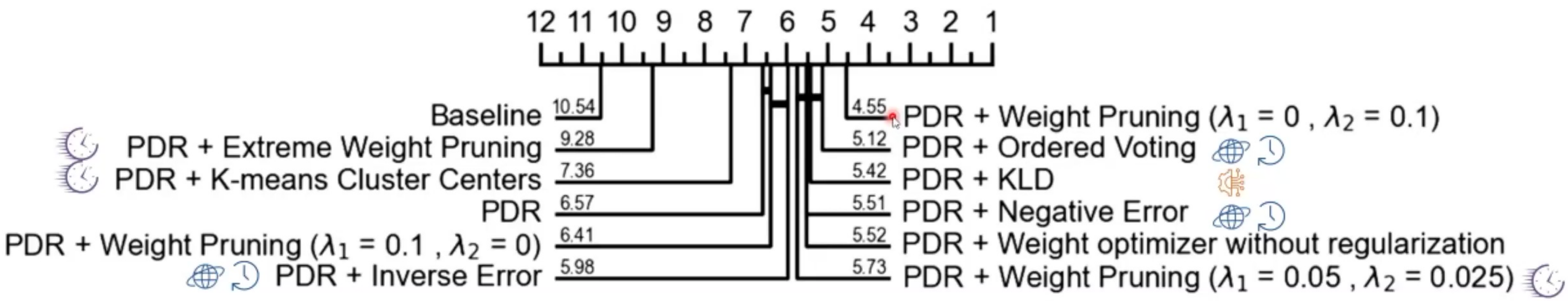


Figure: Comparing the Average Rank of Studied Weighting Methods

- Generalization problem: Decreasing the validation loss $\nRightarrow$ decrease of the test loss
- Weighted PDL with L_2 regularization achieves the best performance in 58% of the datasets.
- Advantage of other weighting:
  - A lower time complexity during training
  - The suitability for online algorithms
  - A speed up of the inference
  - A better generalization

--- FUTURE WORK ---

Extend the PDL Library to:

- Neural Net. Baselines
  With a better transformed input features

- Semi-supervised learning
  current approach: pseudo-label test points, then append to train.
  Lead to 1% improvement
  Any road map to improve the results?

- Uncertainty estimation
  std of ensemble predictions, violation of loop consistency
  "Anchors with a lower uncertainty metric should be weighted higher than anchors with high uncertainty metrics"
  "We believe that other uncertainty metrics that unrelated to the intrinsic consistency metrics might be better suited as for example in Gaussian processes."

--- Results ---

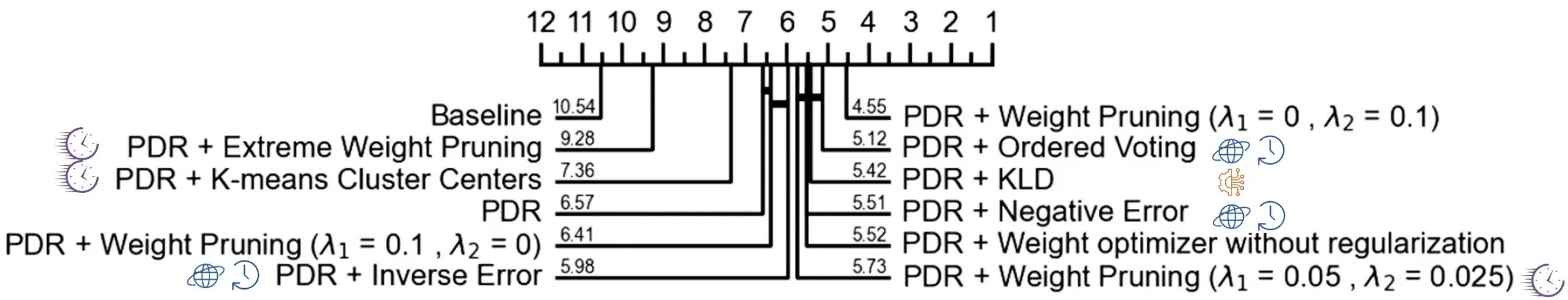


Figure: Comparing the Average Rank of Studied Weighting Methods

- Generalization problem: Decreasing the validation loss $\nRightarrow$ decrease of the test loss
- Weighted PDL with L_2 regularization achieves the best performance in 58% of the datasets.
- Advantage of other weighting:
  - A lower time complexity during training
  - The suitability for online algorithms
  - A speed up of the inference
  - A better generalization