Published by :
https://www.ijert.org/
An International Peer-Reviewed Journal

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026

Stabilizing F1 Decision Thresholds When Positives Are Scarce

A Simple Plug-in Average and an Empirical Comparison with Meta-Learned Thresholds

Debashis Roy*
Department of Computer and System Sciences
Visva-Bharati University (UNESCO World Heritage Centre) Santiniketan, West Bengal, India

Anandarup Roy
Sarojini Naidu College for Women
Dum Dum, Kolkata 700028 West Bengal, India

Utpal Roy
Department of Computer and System Sciences
Visva-Bharati University (UNESCO World Heritage Centre) Santiniketan, West Bengal, India

***Abstract* - Threshold moving is a cheap and effective remedy for class imbalance, but the usual way of choosing the threshold, maximizing F1 on out-of-fold predictions, becomes unreliable when the training set holds only a few dozen positive cases. We compare threshold estimators under controlled positive budgets of 10, 20, 30 and 50 on 31 public imbalanced datasets from 19 families, using logistic regression and random forest classifiers with 20 repetitions per setting (4,840 runs in total). The estimators are the standard cross-validated threshold, the plug-in threshold equal to half the best out-of-fold F1, their equal-weight average, an inverse-variance weighted variant, and two meta-learned estimators trained leave-one-family-out on 16 data-complexity and score meta-features. The equal-weight average needs only the training set, yet it improved mean test F1 over the cross-validated threshold in all eight settings, by 0.005 to 0.010. The gain was significant across datasets in every setting and across dataset families in seven of eight (Holm-adjusted Wilcoxon tests), and the spread of the selected threshold fell by 40 to 50 percent. Meta-learned thresholds added a significant gain only with the random forest at 10 positives (plus 0.008 F1) and otherwise matched or trailed the simple average. We recommend the CV plug-in average as a default whenever positives are scarce.**

Keywords - class imbalance; threshold moving; F1 score; small samples; plug-in classifier; meta-learning; data complexity

I. Introduction

In imbalanced binary classification, the decision threshold applied to a classifier's scores often matters as much as the classifier itself. Threshold moving is cheap, needs no retraining and is widely recommended over resampling [1], [2]. The usual recipe is to choose the threshold that maximizes the target metric, here the F1 score, on out-of-fold (OOF) predictions from cross-validation.

This recipe works when positives are plentiful. When the training set has only 10 to 50 positive cases, it becomes unreliable. The empirical F1 curve is a step function of the threshold, and its maximizer jumps with every positive that enters or leaves a fold. The selected threshold therefore carries high variance, and that variance turns directly into lost test F1. Small-positive settings are common in practice: early-stage medical screening, rare equipment faults, fraud in a new market segment and new software modules with few recorded defects.

Two remedies have been proposed. The first uses theory: for calibrated probabilities, the F1-optimal threshold equals half the best achievable F1 [3], which gives a plug-in threshold. The second uses meta-learning: a model trained on many other datasets predicts the threshold from dataset descriptors [4]. Both are attractive, yet neither has been compared directly with the standard OOF threshold across controlled positive budgets.

This paper makes three contributions:

1) A controlled benchmark of threshold estimators at fixed positive budgets of 10, 20, 30 and 50 on 31 public imbalanced datasets (19 dataset families), with two base classifiers and 20 repetitions per setting.

2) A simple estimator, the *CV–plug-in average*, that averages the OOF threshold with the F1/2 plug-in threshold computed from the same OOF scores. It needs no extra data, no meta-learning and no tuning.

3) An empirical test of whether data-complexity meta-features add value beyond this simple estimator. They largely do not, which is a useful negative result for practitioners considering meta-learned thresholds.

Section II reviews related work. Section III defines the estimators. Section IV describes the experimental protocol, Section V reports results, and Section VI discusses limitations.

II. Related Work

Threshold moving for imbalanced data. Shifting the decision threshold is one of the oldest corrections for class imbalance [5], [6]. It is competitive with resampling such as SMOTE [7] and often simpler [1], [2]. GHOST [8] selects thresholds by resampling the training set, which shares our concern about threshold stability but does not target very small positive counts.

Plug-in theory for the F-measure. For calibrated scores, the F1-optimal decision rule thresholds the conditional probability at half the optimal F1 value [3]. Related consistency results for plug-in classifiers under non-decomposable metrics are given in [9]–[11]. These results describe the population optimum; they do not address how to estimate the threshold when the sample holds only a handful of positives.

Meta-learned thresholds. Zefrehi et al. [4] train a random-forest meta-learner on external datasets to predict the threshold

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026

for rare-positive problems, using meta-features of the imbalance and the score space. Their work motivates our meta-learning comparators. We add a simple label-only baseline and a controlled sweep over the number of positives.

Data complexity. Complexity measures such as feature overlap, neighborhood hardness and class separability describe how difficult a dataset is [12]–[14]. They are mainly used to predict classifier accuracy. We test whether they also help predict the decision threshold.

III. Threshold Estimators

Let the training set contain n cases with k positives. A base classifier is fitted in 5-fold stratified cross-validation, giving an out-of-fold score s_i for every training case. The final classifier is then refitted on all training data and applied to the test set with the chosen threshold τ. All estimators below use only the training set.

Cross-validated threshold (τ_{CV}). The standard choice maximizes the empirical F1 of the OOF scores:

$$\tau_{CV} = arg\ max_{\tau}\ F1(\tau; s, y) \quad (1)$$

The F1 curve is evaluated at every distinct score, and the threshold is set midway between the selected score and the next lower one.

Plug-in threshold (τ_{PI}). If the scores are calibrated probabilities, the F1-optimal threshold equals half the optimal F1 [3]. We plug in the maximum OOF F1, written $\hat{F}^*$:

$$\tau_{PI} = \hat{F}^* / 2 \quad (2)$$

τ_{PI} depends on the OOF scores only through a single summary, so it moves less than τ_{CV} when one positive changes folds. It is biased when scores are poorly calibrated, and $\hat{F}^*$ is optimistic in small samples.

CV–plug-in average (τ_{AVG}). The two estimators fail in different ways: τ_{CV} is noisy but needs no calibration, while τ_{PI} is stable but relies on calibration. We therefore average them with equal weight:

$$\tau_{AVG} = (\tau_{CV} + \tau_{PI}) / 2 \quad (3)$$

Inverse-variance weighted average (τ_{IVW}). As an adaptive variant, the weight is set from B = 50 bootstrap resamples of the OOF pairs. With bootstrap variances V_{CV} and V_{PI} and covariance C, the variance-minimizing weight on τ_{CV} is $w = (V_{PI} - C) / (V_{CV} + V_{PI} - 2C)$, clipped to [0, 1], and $\tau_{IVW} = w \cdot \tau_{CV} + (1 - w) \cdot \tau_{PI}$.

Meta-learned comparators. To test whether external datasets help, we built two estimators in the spirit of [4]. For τ_{META}, a random-forest regressor (300 trees, minimum leaf size 5) predicts the threshold from 16 training-set meta-features: prevalence, log n, log d, log k; OOF AUC, Brier score and $\hat{F}^*$; the mean and standard deviation of the OOF scores per class and overall; kDN for positives and for all cases and N3 [13], [14]; and the maximum Fisher discriminant ratio (log scale). The regression target is the test-optimal threshold on the other datasets. For τ_{SHR}, τ_{CV} is shrunk toward τ_{META} with weight $w = V_{META} / (V_{META} + V_{CV})$, where V_{META} is the meta-learner's mean squared error, estimated per positive budget by grouped cross-validation on the training families only. The meta-learner is always trained leave-one-family-out: the evaluated dataset's family never appears in its training data.

Reference baselines. A fixed threshold of 0.5, and the training prevalence k/n used as the threshold.

IV. Experimental Setup

Datasets. We used the KEEL and imbalanced-learning benchmark collection distributed with the imbalanced_databases Python package [15], [16]. Of its 110 datasets, 107 loaded without error, and 31 have at least 60 positive cases, the number needed to draw a training set with 50 positives while keeping a separate test set. The 31 datasets fall into 19 families (Table I), since several share a source. Dataset sizes range from 214 to 13,086 cases and from 3 to 1,617 features.

TABLE I. Datasets Grouped by Family

Family	Datasets	Positives
vehicle	vehicle0, vehicle1, vehicle2, vehicle3	199–218
yeast	yeast1, yeast3, yeast-0-2-5-6_vs_3-7-8-9, yeast-0-2-5-7-9_vs_3-6-8	99–429
kr-vs-k	one_vs_fifteen, three_vs_eleven, zero-one_vs_draw	78–105
ecoli	ecoli1, ecoli-0_vs_1	77
glass	glass0, glass1	70–76
car	car_good, car-vgood	65–69
NASA defects	KC1, PC1	77–326
12 single-dataset families	ADA, german, haberman, HIVA, page_blocks0, pima, SATIMAGE, segment0, shuttle-c0-vs-c4, sylva, vowel0, wisconsin	81–1029

Protocol. For each dataset and each of 20 repetitions, 30% of the positives and 30% of the negatives were held out as a test set, shared across budgets within that repetition. From the remaining pool, a training set was drawn with exactly k positives (k = 10, 20, 30, 50) and negatives in the dataset's original ratio, capped by availability. The budget k = 50 was skipped for the 3 datasets with fewer than 50 pool positives, leaving 28 datasets. This gives 2,420 training–test runs per base classifier.

Base classifiers. Logistic regression (LR) with standardized features (C = 1) gives fairly calibrated scores. A random forest (RF) with 200 trees gives scores that are typically less calibrated [17]; it tests how robust the plug-in estimators are to the calibration assumption. No separate calibration step was applied, because a calibration fold is itself unreliable with 10 positives.

Evaluation. The primary measure is test F1 at the chosen threshold. We also report the test oracle, the best F1 achievable on that test set with the same scores. The oracle is used only for evaluation and as the meta-learner's training target on other datasets; it is never used to choose a threshold for the dataset being evaluated.

Statistics. Results are first averaged over the 20 repetitions per dataset. Estimators are then compared with the Wilcoxon signed-rank test across datasets. Because datasets in one family are correlated, every test is repeated on family means (19 families). p-values within each budget are Holm-adjusted across the comparisons made [18].

V. Results

The standard cross-validated threshold was the weakest of the data-driven estimators in all eight settings (two classifiers × four budgets). The simple CV–plug-in average improved on it every time, by 0.005 to 0.010 F1.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026

A. Test F1

Table II gives mean test F1. Both fixed rules trail every data-driven estimator by 0.04 to 0.10 F1, so they are not discussed further. The best estimator in each setting closes only 13% to 31% of the gap between τ_{CV} and the test oracle (τ_{AVG}: 12% to 22%). Most of that gap reflects sampling noise in the test set and the small training sample, which no threshold rule can remove.

TABLE II. MEAN TEST F1 OVER DATASETS

Estimator	k = 10	k = 20	k = 30	k = 50
Logistic regression				
Fixed 0.5	0.533	0.571	0.586	0.644
Prevalence	0.556	0.587	0.603	0.649
τ_{CV}	0.602	0.639	0.656	0.698
τ_{PI}	0.603	0.640	0.658	**0.703**
τ_{AVG}	0.609	**0.645**	**0.661**	**0.703**
τ_{IVW}	0.606	0.641	0.659	**0.703**
τ_{META}	0.606	0.633	0.648	0.698
τ_{SHR}	**0.611**	0.643	0.660	0.701
Oracle	0.658	0.682	0.693	0.730
Random forest				
Fixed 0.5	0.566	0.645	0.679	0.698
Prevalence	0.560	0.608	0.632	0.669
τ_{CV}	0.640	0.697	0.726	0.739
τ_{PI}	0.647	0.707	0.730	0.743
τ_{AVG}	0.651	0.707	0.732	**0.744**
τ_{IVW}	0.653	**0.709**	0.731	**0.744**
τ_{META}	**0.659**	**0.709**	**0.734**	**0.744**
τ_{SHR}	0.654	0.707	**0.734**	**0.744**
Oracle	0.700	0.743	0.763	0.769

31 datasets for k ≤ 30 and 28 for k = 50; each value averages 20 repetitions per dataset. Bold: best data-driven estimator in the column. Oracle: best F1 achievable on the test set with the same scores.

B. Paired Comparisons

Table III reports the key paired comparisons at the family level. At the dataset level (31 datasets), τ_{AVG} beat τ_{CV} significantly in all eight settings (Holm-adjusted $p < 0.05$). Three findings follow.

1) *τ_{AVG} beats τ_{CV} consistently.* The difference is significant across families in 7 of 8 settings; the exception is LR at k = 10 ($p = 0.057$).

2) *τ_{AVG} and τ_{PI} are close.* Averaging gives a small further gain over the plug-in alone only for LR at k = 20. Where differences are not significant, τ_{AVG} is never worse on average.

3) *Meta-learning helps only in the most extreme case.* With RF at k = 10, τ_{META} beats τ_{AVG} by 0.008 F1 (27 of 31 datasets, $p = 0.014$). In the other seven settings it brings no significant gain, and with LR it is slightly worse on average.

TABLE III. PAIRED COMPARISONS AT THE FAMILY LEVEL

Comparison	k = 10	k = 20	k = 30	k = 50
$\tau_{AVG} - \tau_{CV}$, LR	+0.007 (0.057)	**+0.005 (0.011)**	**+0.005 (0.023)**	**+0.005 (0.015)**
$\tau_{AVG} - \tau_{CV}$, RF	**+0.010 (0.026)**	**+0.010 (0.005)**	**+0.006 (0.013)**	**+0.004 (0.029)**
$\tau_{AVG} - \tau_{PI}$, LR	+0.006 (0.417)	**+0.005 (0.010)**	+0.003 (0.263)	0.000 (0.772)
$\tau_{AVG} - \tau_{PI}$, RF	+0.003 (0.557)	0.000 (0.841)	+0.002 (0.532)	+0.001 (1.000)
$\tau_{META} - \tau_{AVG}$, LR	−0.003 (0.791)	−0.012 (0.578)	−0.013 (0.311)	−0.005 (0.772)
$\tau_{META} - \tau_{AVG}$, RF	**+0.008 (0.014)**	+0.002 (0.510)	+0.002 (0.532)	0.000 (1.000)
$\tau_{SHR} - \tau_{AVG}$, LR	+0.003 (0.069)	−0.002 (0.999)	−0.001 (0.945)	−0.001 (0.772)
$\tau_{SHR} - \tau_{AVG}$, RF	**+0.004 (0.050)**	0.000 (0.879)	+0.002 (0.532)	0.000 (0.926)

Cell: mean difference in test F1 (first minus second estimator) and, in parentheses, the Holm-adjusted Wilcoxon p-value over 19 family means. Bold: $p < 0.05$.

The adaptive τ_{IVW} placed only 8% to 17% of its weight on τ_{CV} on average, so it behaved almost like τ_{PI}. It was significantly better than τ_{CV} in only 4 of the 8 settings across datasets, against 8 of 8 for τ_{AVG}. The bootstrap variance ignores the bias of τ_{PI}, so a variance-only weight leans too far toward it. Fig. 1 shows the gains over τ_{CV} across budgets.

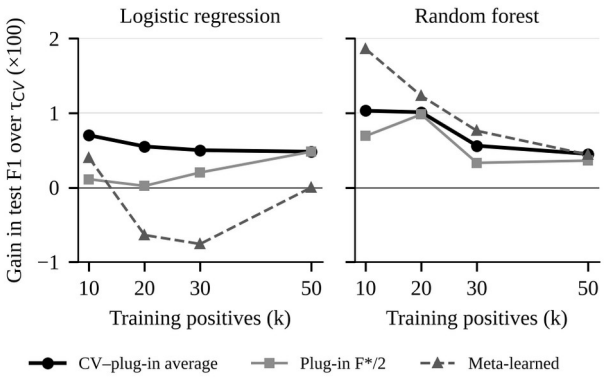


Fig. 1. Mean gain in test F1 over τ_{CV} (×100) by number of training positives. The average gives a steady gain for both classifiers; the meta-learned threshold helps clearly only for the random forest at k = 10.

C. Threshold Stability

The gains come from lower variance. Table IV gives the standard deviation of the selected threshold across the 20 repetitions, averaged over datasets. Averaging cuts the spread of the threshold by about 40% to 50%. τ_{PI} and τ_{META} are even more stable, but their lower variance is partly offset by bias, so they do not dominate τ_{AVG} in F1.

TABLE IV. MEAN WITHIN-DATASET STANDARD DEVIATION OF THE SELECTED THRESHOLD

Estimator	LR, k = 10	LR, k = 50	RF, k = 10	RF, k = 50
τ_{CV}	0.121	0.093	0.108	0.072
τ_{AVG}	0.070	0.050	0.065	0.038
τ_{PI}	0.041	0.016	0.041	0.015
τ_{META}	0.035	0.015	0.023	0.016

VI. DISCUSSION

Why averaging works. τ_{CV} and τ_{PI} are computed from the same OOF scores but react to noise differently. τ_{CV} follows the exact location of the F1 peak, which jumps when one positive changes rank; τ_{PI} follows only the peak's height, which changes more smoothly. τ_{PI} is biased, though: $\hat{F}^*$ is optimistic in small samples, and the F1/2 rule holds only for calibrated scores. An equal-weight average reduces the variance of τ_{CV} while diluting the bias of τ_{PI}. Because both errors are present, the fixed weight of one half beat the variance-only weight of τ_{IVW}, which ignored the bias.

Calibration. The plug-in rule assumes calibrated scores, yet τ_{PI} and τ_{AVG} helped at least as much with the random forest as with logistic regression. A likely reason is that both $\hat{F}^*$ and the oracle threshold are measured on the same, equally

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026

miscalibrated score scale, so part of the miscalibration cancels. We did not test strongly miscalibrated models such as unregularized boosted trees or SVM margins.

When meta-learning is worth it. Data-complexity meta-features brought a significant gain only with the random forest at 10 positives, by about 0.008 F1. Elsewhere they matched or trailed the simple average, while needing a corpus of external datasets, a complexity-measure library and a trained meta-model. For most practitioners, τ_{AVG} gives most of the available benefit at no extra cost. Meta-learning may pay off when positives are extremely scarce and a large, relevant corpus of past datasets exists.

Practical recommendation. When the training set holds fewer than about 50 positives, report $\tau_{AVG} = (\tau_{CV} + \hat{F}^*/2) / 2$ instead of the raw OOF argmax. It costs one line of code.

Limitations. First, only F1 was studied, because the plug-in rule is specific to it; other F-beta measures have analogous rules but were not tested. Second, the benchmark has 31 datasets from 19 families, mostly small tabular problems from KEEL, and results may differ for high-dimensional text or image data. Third, two base classifiers were used, without explicit calibration. Fourth, the absolute gains are small (0.005 to 0.010 F1); they are consistent across datasets, but their practical value depends on the application. Fifth, the training sets keep the original class ratio, so a budget of k positives also implies few negatives on low-imbalance datasets.

VII. CONCLUSION

With 10 to 50 positive training cases, the usual cross-validated F1 threshold is noisy enough to cost measurable test performance. Averaging it with the plug-in threshold $\hat{F}^*/2$, computed from the same out-of-fold scores, raised test F1 in all eight settings studied, by 0.005 to 0.010 on average, and cut the spread of the threshold by 40% to 50%. Meta-learned thresholds from data-complexity features gave a further gain only in the scarcest setting with a random forest. A simple, training-set-only correction is therefore a sensible default before reaching for meta-learning. Future work should test other F-beta measures, explicitly calibrated and poorly calibrated models, and larger, more diverse dataset collections.

DECLARATIONS

Ethical approval. Not applicable. The study uses only existing, de-identified datasets from public repositories. No new data were collected from humans or animals.

Consent to participate. Not applicable. No human participants were recruited.

Consent to publish. Not applicable. The manuscript contains no individual person's data, images, or identifying information.

Competing interests. The authors have no competing interests to declare that are relevant to the content of this article.

Funding. No funding.

Data availability. The datasets analysed in this study are available from the public repositories listed.

Use of AI-assisted tools. During the preparation of this manuscript, the authors used Claude (Anthropic) to help develop the study design, write the experiment code, draft the text, and improve English language, grammar, and readability. The authors reviewed and edited all AI-assisted content and take full responsibility for the final content of the manuscript.

REFERENCES

[1] H. He and E. A. Garcia, "Learning from imbalanced data," IEEE Trans. Knowl. Data Eng., vol. 21, no. 9, pp. 1263–1284, 2009.

[2] G. Collell, D. Prelec, and K. R. Patil, "A simple plug-in bagging ensemble based on threshold-moving for classifying binary and multiclass imbalanced data," Neurocomputing, vol. 275, pp. 330–340, 2018.

[3] Z. C. Lipton, C. Elkan, and B. Naryanaswamy, "Optimal thresholding of classifiers to maximize F1 measure," in Machine Learning and Knowledge Discovery in Databases (ECML PKDD 2014), LNCS, vol. 8725, pp. 225–239, 2014, doi: 10.1007/978-3-662-44851-9_15.

[4] H. G. Zefrehi, G. Sheikhi, and H. Altınçay, "Threshold prediction for detecting rare positive samples using a meta-learner," Pattern Anal. Appl., vol. 26, no. 1, pp. 289–306, 2023, doi: 10.1007/s10044-022-01103-1.

[5] F. Provost, "Machine learning from imbalanced data sets 101," in Proc. AAAI Workshop on Learning from Imbalanced Data Sets, Tech. Rep. WS-00-05, 2000.

[6] C. Elkan, "The foundations of cost-sensitive learning," in Proc. 17th Int. Joint Conf. Artificial Intelligence (IJCAI), 2001, pp. 973–978.

[7] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique," J. Artif. Intell. Res., vol. 16, pp. 321–357, 2002.

[8] C. Esposito, G. A. Landrum, N. Schneider, N. Stiefl, and S. Riniker, "GHOST: adjusting the decision threshold to handle imbalanced data in machine learning," J. Chem. Inf. Model., vol. 61, no. 6, pp. 2623–2640, 2021.

[9] M.-J. Zhao, N. Edakunni, A. Pocock, and G. Brown, "Beyond Fano's inequality: bounds on the optimal F-score, BER, and cost-sensitive risk and their implications," J. Mach. Learn. Res., vol. 14, pp. 1033–1090, 2013.

[10] O. Koyejo, N. Natarajan, P. Ravikumar, and I. S. Dhillon, "Consistent binary classification with generalized performance metrics," in Advances in Neural Information Processing Systems 27, 2014.

[11] H. Narasimhan, R. Vaish, and S. Agarwal, "On the statistical consistency of plug-in classifiers for non-decomposable performance measures," in Advances in Neural Information Processing Systems 27, 2014.

[12] T. K. Ho and M. Basu, "Complexity measures of supervised classification problems," IEEE Trans. Pattern Anal. Mach. Intell., vol. 24, no. 3, pp. 289–300, 2002.

[13] M. R. Smith, T. Martinez, and C. Giraud-Carrier, "An instance level analysis of data complexity," Mach. Learn., vol. 95, no. 2, pp. 225–256, 2014.

[14] A. C. Lorena, L. P. F. Garcia, J. Lehmann, M. C. P. Souto, and T. K. Ho, "How complex is your classification problem? A survey on measuring classification complexity," ACM Comput. Surv., vol. 52, no. 5, art. 107, 2019.

[15] J. Alcalá-Fdez, A. Fernández, J. Luengo, J. Derrac, S. García, L. Sánchez, and F. Herrera, "KEEL data-mining software tool: data set repository, integration of algorithms and experimental analysis framework," J. Mult.-Valued Log. Soft Comput., vol. 17, no. 2–3, pp. 255–287, 2011.

[16] G. Kovács, "An empirical comparison and evaluation of minority oversampling techniques on a large number of imbalanced datasets," Appl. Soft Comput., vol. 83, art. 105662, 2019.

[17] A. Niculescu-Mizil and R. Caruana, "Predicting good probabilities with supervised learning," in Proc. 22nd Int. Conf. Machine Learning (ICML), 2005, pp. 625–632.

[18] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," J. Mach. Learn. Res., vol. 7, pp. 1–30, 2006.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)