Covariance Shrinkage in Linear Discriminant Analysis: A Protocol-Bounded Empirical Comparison

Anonymous

Venue-neutral anonymous research report; no specific venue or submission rule was supplied.

Abstract

Covariance shrinkage is often used to stabilize linear discriminant analysis (LDA), but the practical effect depends on the data regime, implementation, and evaluation protocol. We compare matched unregularized LDA and LDA with scikit-learn's automatic shrinkage on the processed handwritten-digit benchmark, using fixed-feature preprocessing and a fixed logistic-regression reference. A stratified 20% holdout was locked before development analysis. On the remaining 1,437 examples, each method was evaluated on the same 25 repeated stratified outer splits at four nested training fractions; accuracy and macro-F1 were summarized descriptively with sample standard deviation and interquartile range. Automatic shrinkage improved mean development accuracy over unregularized LDA by 1.14 percentage points at the smallest fraction and by 0.24 points when the full outer training folds were used, while paired differences included both wins and losses. On the single authorized 360-example holdout, unregularized LDA reached accuracy 0.9583 and macro-F1 0.9585, shrinkage LDA reached 0.9639 and 0.9640, and logistic regression reached 0.9750 and 0.9752. The two LDA variants differed on only two holdout predictions: shrinkage corrected two examples and lost none. These results support a small, protocol-specific benefit for this benchmark, not a general claim that shrinkage is more accurate or more stable. The repeated folds are correlated, the final holdout was evaluated once, and no independent dataset, mechanism analysis, tuning study, or deployment evaluation is available.

1 Introduction

Linear discriminant analysis (LDA) is a classical approach to multiclass discrimination that represents classes through their means and a shared covariance structure [Fisher, 1936]. In moderate- or high-dimensional settings, an empirical covariance matrix can be poorly conditioned relative to the amount of data available for each class. Shrinkage estimators address this problem by pulling covariance estimates toward a structured target; the Ledoit–Wolf line of work is a prominent example of this strategy [Ledoit and Wolf, 2004]. Modern software exposes these choices through a common estimator interface, which makes the implementation details part of the empirical question rather than an incidental coding choice.

This report asks a narrow question: on the scikit-learn handwritten-digit benchmark, under a prespecified stratified split and matched resampling scheme, does automatic covariance shrinkage change the predictive accuracy and resampling variation of a linear discriminant classifier relative to the same classifier without shrinkage? A fixed L2-regularized multiclass logistic regression is included as a reference point, not as a tuned competitor or a claim about the best available classifier. The analysis is intentionally bounded. It studies one processed dataset, one software implementation, one fixed configuration per method, and one authorized final holdout evaluation.

The report makes four contributions of scope and evidence rather than novelty. First, it records an implementation-faithful comparison in which preprocessing, outer folds, training fractions, and

evaluation metrics are matched. Second, it reports paired differences between the two LDA variants so that a change in the mean is not confused with a uniform per-split improvement. Third, it separates the development resampling analysis from the locked final holdout and examines class-level diagnostics on that holdout. Fourth, it makes the uncertainty and reproducibility boundaries explicit: the development summaries are descriptive, the repeated splits overlap, and the one-batch holdout does not provide an estimate of future stability.

2 Related context

2.1 Discriminant analysis and covariance regularization

Fisher's formulation established the use of multiple measurements for separating groups through linear discriminant functions [Fisher, 1936]. In the implementation studied here, the discriminant coefficients are obtained from a shared covariance estimate and class means. The comparison therefore changes one structural ingredient—the covariance estimate—while holding the outer feature transformation and classifier family fixed. The shrinkage variant uses the software's automatic Ledoit–Wolf path, whose bibliographic provenance is reported by Ledoit and Wolf [2004]; this report does not make a theory-level claim about optimality or about the shrinkage intensity outside the measured protocol.

2.2 Implementation and evaluation context

The estimators and dataset are from scikit-learn, an established Python machine-learning library [Pedregosa et al., 2011]. The dataset documentation describes 1,797 examples of 8-by-8 digit images represented by 64 integer-valued features and ten classes [scikit-learn developers, 2024a]. Because a library option such as `shrinkage=auto` can include internal preprocessing and a particular linear solver, we inspected the versioned scikit-learn 1.5.1 source for the covariance and solver functions used by the saved analysis [scikit-learn developers, 2024b,c]. This inspection motivates the explicit method description in Section 4; it is not evidence of predictive performance.

Repeated cross-validation can be useful for describing how results vary over analysis splits, but its folds are not independent replications. In particular, overlap creates dependence that complicates variance estimation and makes a naive standard error unsuitable as a strict confidence interval [Bengio and Grandvalet, 2004]. We therefore report sample standard deviation, interquartile range, and paired win/tie/loss counts as descriptive diagnostics. They quantify variation in the recorded resamples without assigning a formal inferential interpretation that the design cannot support.

3 Data and evaluation protocol

3.1 Dataset and split

We use `load_digits`, a processed image-classification benchmark with $N = 1{,}797$ observations, $p = 64$ features, and $K = 10$ digit classes. Each observation is an 8-by-8 image flattened to integer-valued features in the range 0–16 [scikit-learn developers, 2024a]. The data were split once with stratified sampling into a development set of 1,437 observations (80%) and a final holdout of 360 observations (20%), using random seed 20260908. The holdout labels and features were not opened by the development resampling analysis. The final evaluation was authorized and completed exactly once after development was complete; it was not used for method selection or tuning.

3.2 Matched development resampling

Development performance was measured with a 5-fold, 5-repeat stratified outer scheme, for 25 outer resamples in total. All methods used the same outer train/test partitions. Within each outer training fold, without-replacement prefixes were formed at fractions 0.25, 0.50, 0.75, and 1.00, with class quotas allocated by largest fractional remainder and ties resolved by ascending class label. Thus, the four points at a fraction are not four independent datasets, and the repeated outer folds are correlated. Each method was fitted and evaluated separately on each resulting cell, for $25 \times 4 \times 3 = 300$ development fits/evaluations. The saved development record contains zero fit failures and zero recorded warnings.

The primary metric was accuracy, the proportion of held-out examples whose predicted label equals the true label. Macro-F1 was secondary: the F1 score was computed for each of the ten classes and then averaged with equal class weight. For each method and fraction, we report the mean, sample standard deviation (SD), and interquartile range (IQR) over the 25 resamples. For shrinkage versus unregularized LDA, we also report the signed paired difference on each shared outer split, the mean SD and IQR of those differences, and descriptive win/tie/loss counts. A tie uses the recorded tolerance of 10^{-12}. No p-values, strict confidence intervals, or independence-based standard errors are reported.

3.3 Final evaluation and controls

After the development record was complete, each fixed method was fitted once on the full development set and predicted once on the locked 360-example holdout. The final record contains no warnings or failures. The computation was sequentially thread-limited to two threads. The recorded environment used Python 3.12.7, NumPy 1.26.4, SciPy 1.13.1, scikit-learn 1.5.1, joblib 1.4.2, and threadpoolctl 3.5.0 on arm64 Darwin. These controls document the execution context; they do not turn one holdout batch into an uncertainty estimate.

4 Methods

4.1 Common feature construction

For one fit, let $X \in \mathbb{R}^{n \times p}$ be the training design matrix, with row $X_{i:} = x_i^\top$ for $x_i \in \mathbb{R}^p$, and let $y_i \in \{0, \ldots, K-1\}$. The pipeline first fits a `StandardScaler` on the training rows only. With training mean $\mu \in \mathbb{R}^p$ and scale vector $s \in \mathbb{R}^p$, the transformed matrix $Z \in \mathbb{R}^{n \times p}$ has entries

$$z_{ij} = \frac{x_{ij} - \mu_j}{s_j}. \tag{1}$$

The same fitted transformation is applied to the corresponding evaluation rows. This pipeline placement prevents the evaluation rows from contributing to feature means or scales.

4.2 Unregularized and shrinkage LDA

Both LDA variants use the `lsqr` solver. Let $M \in \mathbb{R}^{K \times p}$ contain class means as rows, $M_{k:} = \hat{\mu}_k^\top$, and let $S_k \in \mathbb{R}^{p \times p}$ be the covariance block for class k. With empirical class priors $\hat{\pi}_k$, the shared covariance has the dimensionally valid form

$$S = \sum_{k=1}^{K} \hat{\pi}_k S_k \in \mathbb{R}^{p \times p}. \tag{2}$$

For the unregularized model, each S_k follows the empirical covariance path. For the shrinkage model, each class block follows the implementation's automatic shrinkage path. A useful schematic for the latter is

$$S_k^{\mathrm{sh}} = (1-\lambda_k)S_k + \lambda_k \frac{\mathrm{tr}(S_k)}{p} I_p, \qquad S_k^{\mathrm{sh}} \in \mathbb{R}^{p\times p}, \tag{3}$$

but Equation 3 is a representation of the shrinkage form, not a replacement for the estimator executed in this study. In scikit-learn 1.5.1, `shrinkage=auto` computes class-specific covariance blocks through its internal standardization and Ledoit–Wolf routine and then combines the blocks with class priors. The outer `StandardScaler` and this internal class-block operation are distinct.

The implementation does not form an explicit inverse. Instead, after constructing S, it solves the least-squares system

$$SC = M^\top, \qquad S \in \mathbb{R}^{p\times p}, \quad M^\top \in \mathbb{R}^{p\times K}, \quad C \in \mathbb{R}^{p\times K}, \tag{4}$$

and stores $B = C^\top \in \mathbb{R}^{K\times p}$, so the coefficient for class k is the row vector $\beta_k^\top = B_{k:}$. The class score for a transformed row $z \in \mathbb{R}^p$ is

$$g_k(z) = \beta_k^\top z + b_k, \qquad b_k = -\frac{1}{2}\hat{\mu}_k^\top \beta_k + \log \hat{\pi}_k, \tag{5}$$

where $\beta_k \in \mathbb{R}^p$ is the transpose of the stored coefficient row. Each product in Equation 5 is scalar, and prediction selects the class with the largest score. The dimensions and row/column orientations in Equations 2–5 match the versioned solver path used for the saved results.

4.3 Logistic reference

The reference is multiclass logistic regression with an L2 penalty, $C = 1.0$, the `lbfgs` solver, maximum 2,000 iterations, and random state 20260908. It uses the same training-fold-only standardization pipeline. Its class probabilities have the softmax form

$$p(y = k \mid z) = \frac{\exp(\theta_k^\top z + a_k)}{\sum_{\ell=1}^{K} \exp(\theta_\ell^\top z + a_\ell)}, \tag{6}$$

and the predicted class is the maximum-probability class. The fitted objective is the implementation's multiclass log-loss with L2 regularization; C is the inverse regularization-strength parameter. This fixed reference was not tuned and is included to contextualize the two LDA variants, not to define a model-selection contest.

5 Development results

Figure 1 shows the saved mean accuracy and macro-F1 over the 25 matched resamples at each nested training fraction. Automatic shrinkage was above unregularized LDA in mean accuracy at all four fractions: 0.9294 versus 0.9180 at 25%, 0.9429 versus 0.9395 at 50%, 0.9489 versus 0.9456 at 75%, and 0.9502 versus 0.9478 at 100%. The corresponding macro-F1 means were 0.9293 versus 0.9181, 0.9430 versus 0.9396, 0.9490 versus 0.9457, and 0.9503 versus 0.9480. The mean difference therefore decreased as more training data were used, from 1.14 percentage points in accuracy at 25% to 0.24 points at 100%. Logistic regression was higher than both LDA variants at every displayed fraction, with 0.9662 accuracy and 0.9661 macro-F1 at the full outer training fold.

The whiskers in Figure 1 are sample SD across the 25 resamples, not confidence intervals. At 25%, for example, the accuracy SD was 0.0139 for unregularized LDA and 0.0160 for shrinkage

LDA; at 100%, it was 0.0111 and 0.0096, respectively. Thus the mean improvement does not imply a uniformly smaller spread. Shrinkage's accuracy IQR was smaller at 50%, 75%, and 100% (0.0106, 0.0105, and 0.0070 versus 0.0139, 0.0174, and 0.0139), but larger at 25% (0.0211 versus 0.0107). These are descriptive features of the saved splits.

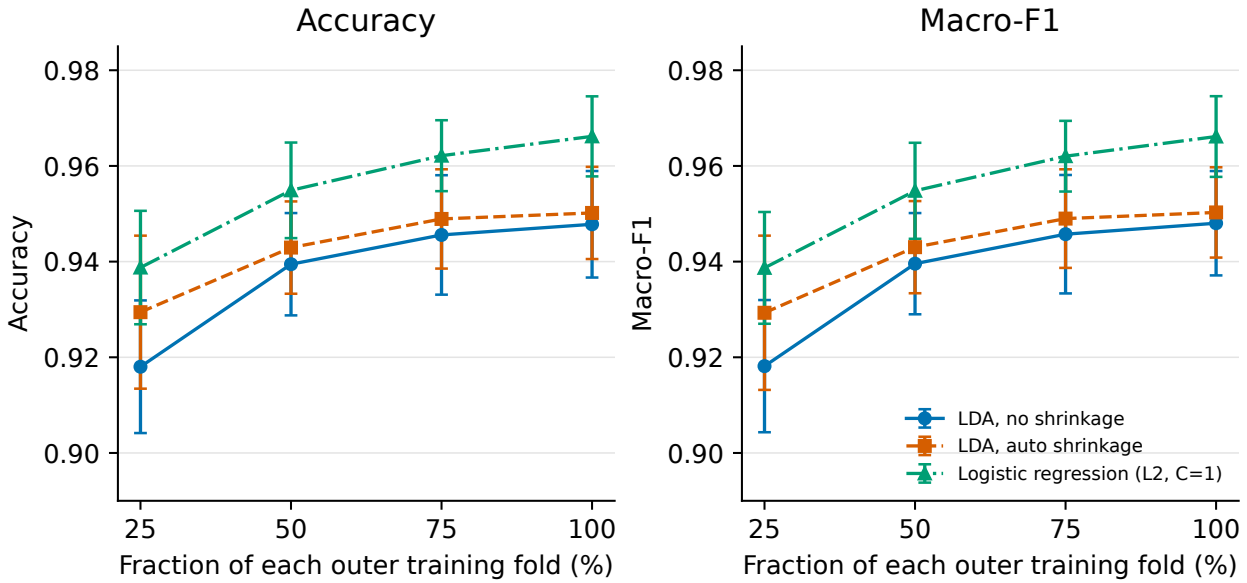

Figure 1: Development comparison across four nested fractions of each outer training fold. Points are saved means and whiskers are sample SD over $n = 25$ matched resamples per fraction; the display is descriptive and does not represent a confidence interval. Accuracy and macro-F1 are unitless proportions. Source: frozen development summary and fold records; generated by the reproducible plotting script.

Figure 2 makes the paired structure explicit by plotting all 25 shrinkage-minus-unregularized differences at each fraction. The accuracy mean differences were 0.0114, 0.0035, 0.0033, and 0.0024 at 25%, 50%, 75%, and 100%, respectively. The paired W/T/L counts were 15/5/5, 13/5/7, 15/3/7, and 12/7/6. For macro-F1, the corresponding mean differences were 0.0112, 0.0035, 0.0033, and 0.0023, with W/T/L counts 18/0/7, 15/0/10, 16/0/9, and 16/0/9. Both positive and negative paired observations appear, including at the full fraction. The matched design makes these differences more informative than a comparison of unrelated aggregate means, while their dependence prevents treating the 25 points as independent replications.

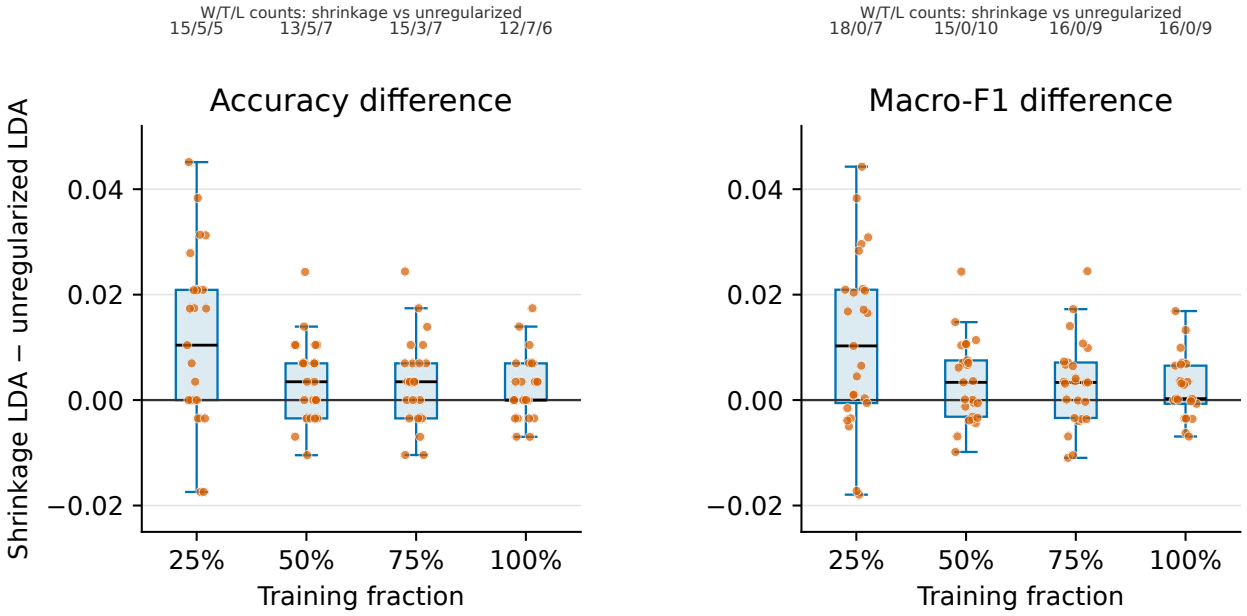

Figure 2: Paired development differences for automatic-shrinkage LDA minus unregularized LDA. Each point is one of the $n = 25$ shared outer resamples at the indicated fraction; boxes summarize the same observations, the horizontal line marks zero, and the separate annotation strip above each data axis gives descriptive wins/ties/losses. No inferential interval is implied. Source: frozen fold-level records and paired summaries; generated by the reproducible plotting script.

The class summaries also show heterogeneity rather than a uniform class-wise shift. For example, shrinkage increased the mean recall of digit 8 relative to unregularized LDA at every fraction, whereas the direction for digit 3 changed from positive at 25% to negative at the three larger fractions; digit 9 was negative at the larger fractions. These class-level aggregates are not a mechanism analysis. They only indicate that the aggregate benefit is composed of class-specific changes whose direction can depend on the training fraction.

6 Final holdout results

Table 1 reports the single authorized evaluation on the 360-example holdout. Unregularized LDA obtained 345 correct predictions (accuracy 0.958333; macro-F1 0.958511), while automatic-shrinkage LDA obtained 347 correct predictions (0.963889; 0.964025). Logistic regression obtained 351 correct predictions (0.975000; 0.975209). These point estimates are consistent with the direction observed in development, but the final batch supplies no error bars and was not repeated.

Table 1: Final holdout metrics from one fit and one prediction batch per fixed method ($n = 360$). Values are point estimates; "errors" is 360−correct.

Method	Accuracy	Correct	Errors	Macro-F1
LDA, no shrinkage	0.958333	345	15	0.958511
LDA, auto shrinkage	0.963889	347	13	0.964025
Logistic regression (L2, $C = 1$)	0.975000	351	9	0.975209

The same-example paired diagnostic is more granular than the aggregate difference. Both LDA variants were correct on 345 examples and incorrect on 13; shrinkage was correct where unregularized LDA was not on two examples, and the reverse transition occurred zero times. The resulting accuracy difference is exactly $2/360 = 0.005556$. Figure 3 displays these counts alongside the final point estimates. This comparison isolates a small number of changed decisions; it does not establish that shrinkage would win on another holdout.

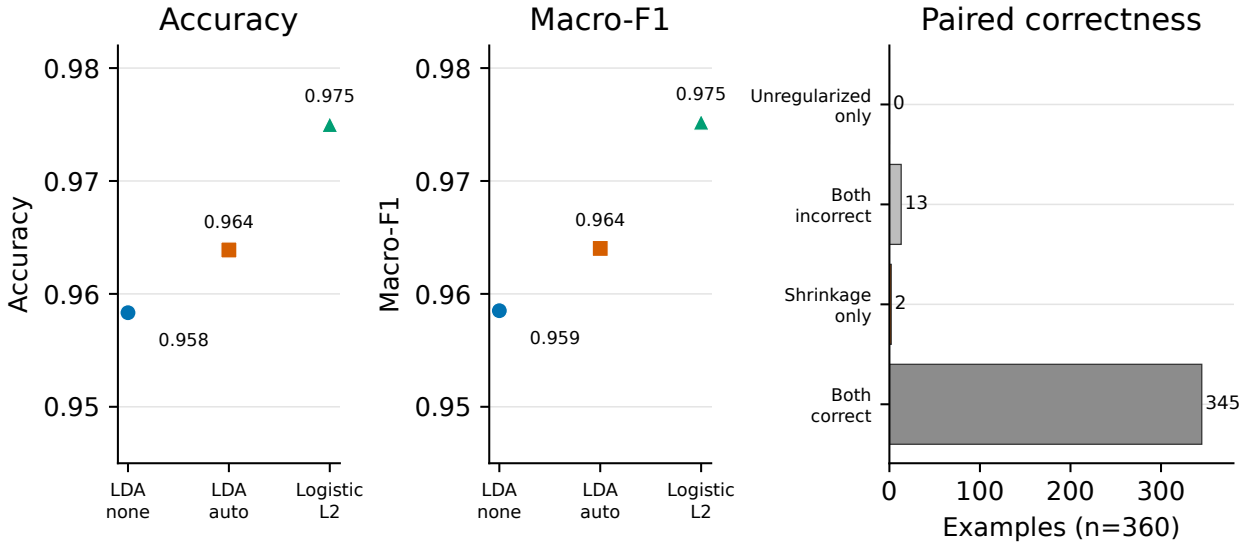

Figure 3: Final one-batch point estimates and same-example correctness transitions for the two LDA variants. The metric panels show saved accuracy and macro-F1 on the locked $n = 360$ holdout; the transition panel uses a zero-based bar axis and counts both-correct, shrinkage-only-correct, both-incorrect, and unregularized-only-correct examples. No uncertainty estimate is claimed. Source: frozen final summary, method records, and paired correctness record; generated by the reproducible plotting script.

Figure 4 shows final per-digit recall and F1. Relative to unregularized LDA, shrinkage recall increased for digits 3 and 8 and was equal for the other eight digits. F1 increased for digits 3, 8, and 9; the digit-9 recall values were equal, so its F1 change is precision-driven. The affected confusion rows provide a concrete accounting: for true digit 3, shrinkage changed one prediction from digit 8 to digit 3; for true digit 8, it changed one prediction from digit 9 to digit 8; and for digit 9, the number of true positives was unchanged while one false positive from true digit 8 disappeared. These statements describe the observed label transitions, not why the covariance estimate caused them.

7 Discussion and limitations

The answer to the prespecified question is conditional. In this benchmark and protocol, automatic shrinkage gave a modest positive mean difference over unregularized LDA across all four development fractions and was two examples better on the single final holdout, while a fixed logistic reference was higher than both. The shrinkage advantage narrowed as the training fraction grew, included losses on individual development splits, and did not uniformly reduce variability. The evidence therefore supports a protocol-specific empirical association between the automatic shrinkage option and slightly higher observed scores, not a universal accuracy or stability advantage.

Several limitations constrain interpretation.

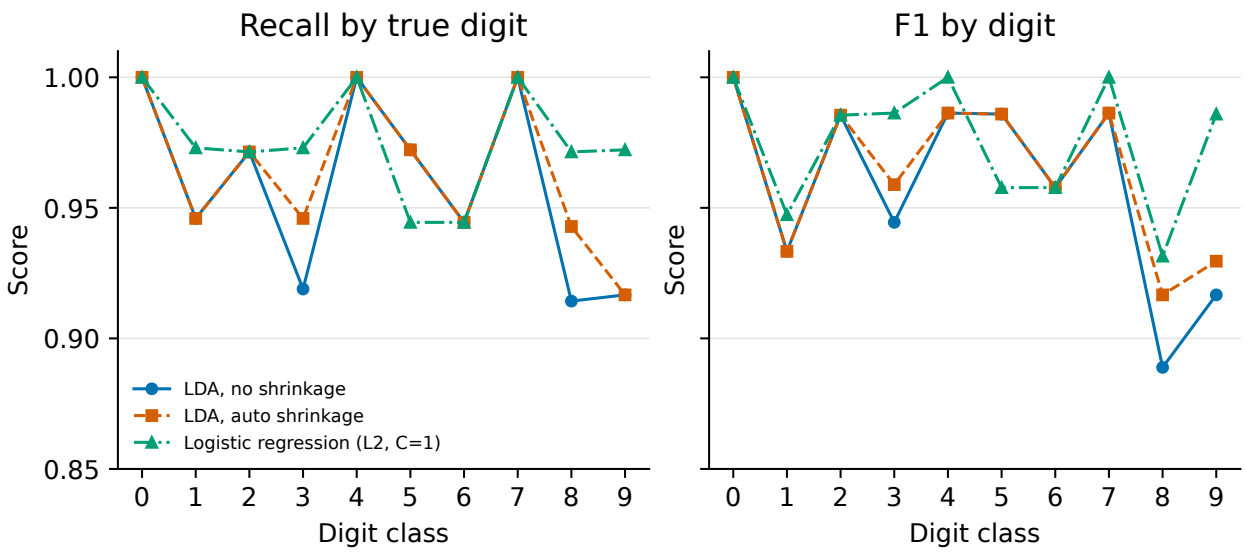

Figure 4: Per-class final-holdout diagnostics for all ten digit classes and all three fixed methods. Points are recall and F1 point estimates on the same $n = 360$ locked holdout; no class-level uncertainty model is claimed. Source: frozen final method records; generated by the reproducible plotting script.

Data scope. The benchmark is a processed, low-resolution digit dataset. It does not provide the writer-disjoint structure or distributional breadth needed to assess generalization across writers, acquisition conditions, or deployment populations. The result should not be extrapolated to other p/n regimes, class structures, or covariance geometries.

Resampling dependence and final uncertainty. The 25 development splits share observations and are repeated versions of the same finite dataset. Their SD and IQR are useful descriptions of the recorded protocol, but they are not independent-replication uncertainty. The final holdout was opened and scored once, so its two-example difference cannot be converted here into a reliable estimate of future win probability.

Fixed configurations. The logistic reference used one prespecified C and solver configuration, and the LDA comparison used the two saved covariance options. There was no hyperparameter search, alternative preprocessing study, solver sensitivity analysis, or independent replication. Consequently, the comparison says what these configurations did, not which model family is optimal.

Mechanism and source coverage. The class-level transitions identify where predictions changed but do not establish a causal mechanism. The implementation semantics were checked against versioned scikit-learn source. The public bibliographic record for Ledoit–Wolf was verified, but the full text was not available in the recorded source pass; likewise, a lead concerning Mkhadri was not used because its full text was inaccessible. No detailed optimality or theoretical claim from those access-limited sources is made.

Operational history. The initial environment setup encountered an offline dependency-install failure and recovered using the recorded local environment; this was an operational event, not a performance result. A separate early manuscript audit assumed an incorrect prediction-record

key and failed before any model computation; the corrected audit used the recorded key. These events remain in the audit record so that the clean final metrics are not confused with a claim of an uninterrupted workflow.

8 Reproducibility and declarations

The experimental protocol is determined by the dataset version/API, stratified split seed 20260908, 80/20 development–holdout boundary, 5 repeats of 5 outer folds, four nested fractions, the three fixed estimator configurations, and the package versions reported in Section 3. The final evaluation consisted of one fit and one prediction batch per method after development completion. The four manuscript figures are generated by a Python script that reads the saved metric records and performs no fitting, scoring, resampling, or bootstrap sampling. The exact rebuilding commands and artifact checks are provided in the accompanying build and QA records.

This anonymous draft does not supply author names, affiliations, funding information, competing-interest declarations, ethics or consent determinations, or a final data/code availability statement. Those declarations await author confirmation and venue selection; their absence here is not evidence that the underlying declarations are negative or unnecessary. This draft was prepared with assistance from an AI research agent for evidence-bounded writing, figure scripting, and QA; human authors remain responsible for verification, authorship, and all final scientific and submission decisions. Human verification of the evidence mapping, authorship, declarations, visual output, and any eventual submission decision remains pending. The procedural research-agent workflow used to package the evidence and audit trail is described by Kassis et al. [2026]; that workflow does not constitute scientific or submission approval.

A Detailed protocol and summary tables

Table 2 gives the fixed estimator settings. Every estimator was wrapped with the same training-only standardization operation. No method was selected using the final holdout.

Table 2: Fixed estimator configurations used in the saved evaluation.

Method	Configuration
LDA, no shrinkage	`LinearDiscriminantAnalysis(solver=lsqr, shrinkage=None)` with outer `StandardScaler`.
LDA, auto shrinkage	`LinearDiscriminantAnalysis(solver=lsqr, shrinkage=auto)` with the same outer `StandardScaler`; the auto path includes the version-specific internal covariance operations described in Section 4.
Logistic reference	`LogisticRegression(penalty=l2, C=1.0, solver=lbfgs, max_iter=2000, random_state=20260908)` with the same outer `StandardScaler`.

Tables 3 and 4 report the full saved development summaries. Values are rounded to four decimals for readability; SD is the sample SD and IQR is the 75th–25th percentile range over 25 matched resamples.

Table 3: Development accuracy summaries ($n = 25$ resamples per fraction).

Fraction	Method	Mean	SD	IQR
25%	LDA, no shrinkage	0.9180	0.0139	0.0107
25%	LDA, auto shrinkage	0.9294	0.0160	0.0211
25%	Logistic L2	0.9388	0.0119	0.0207
50%	LDA, no shrinkage	0.9395	0.0107	0.0139
50%	LDA, auto shrinkage	0.9429	0.0096	0.0106
50%	Logistic L2	0.9549	0.0100	0.0139
75%	LDA, no shrinkage	0.9456	0.0125	0.0174
75%	LDA, auto shrinkage	0.9489	0.0104	0.0105
75%	Logistic L2	0.9621	0.0074	0.0140
100%	LDA, no shrinkage	0.9478	0.0111	0.0139
100%	LDA, auto shrinkage	0.9502	0.0096	0.0070
100%	Logistic L2	0.9662	0.0084	0.0070

Table 4: Development macro-F1 summaries ($n = 25$ resamples per fraction).

Fraction	Method	Mean	SD	IQR
25%	LDA, no shrinkage	0.9181	0.0138	0.0124
25%	LDA, auto shrinkage	0.9293	0.0161	0.0206
25%	Logistic L2	0.9387	0.0117	0.0197
50%	LDA, no shrinkage	0.9396	0.0106	0.0141
50%	LDA, auto shrinkage	0.9430	0.0096	0.0110
50%	Logistic L2	0.9548	0.0100	0.0135
75%	LDA, no shrinkage	0.9457	0.0124	0.0176
75%	LDA, auto shrinkage	0.9490	0.0103	0.0098
75%	Logistic L2	0.9620	0.0074	0.0135
100%	LDA, no shrinkage	0.9480	0.0109	0.0134
100%	LDA, auto shrinkage	0.9503	0.0094	0.0073
100%	Logistic L2	0.9661	0.0084	0.0071

Table 5: Paired development differences for automatic-shrinkage LDA minus unregularized LDA. W/T/L counts are based on the saved 10^{-12} tie tolerance.

	Accuracy						Macro-F1					
Fraction	Mean	SD	IQR	W	T	L	Mean	SD	IQR	W	T	L
25%	0.0114	0.0164	0.0209	15	5	5	0.0112	0.0163	0.0215	18	0	7
50%	0.0035	0.0076	0.0104	13	5	7	0.0035	0.0078	0.0107	15	0	10
75%	0.0033	0.0082	0.0104	15	3	7	0.0033	0.0082	0.0105	16	0	9
100%	0.0024	0.0060	0.0070	12	7	6	0.0023	0.0058	0.0072	16	0	9

B Confusion changes for the final LDA comparison

For completeness, the only rows that changed between the two LDA confusion matrices are shown below. Rows are true classes and columns are predicted classes; omitted rows are identical between variants. These counts are the source of the class-level statements in Section 6.

Table 6: Affected final-holdout confusion-matrix rows for unregularized versus automatic-shrinkage LDA.

	Unregularized LDA										Auto-shrinkage LDA									
True	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
3	0	0	0	34	0	0	0	0	3	0	0	0	0	35	0	0	0	0	2	0
8	0	2	0	0	0	0	0	0	32	1	0	2	0	0	0	0	0	0	33	0

The digit-9 F1 increase is not caused by a change in its own recall: both variants had 33 true positives out of the same 36 true digit-9 examples. It follows from a reduction in false positives for predicted digit 9, including the disappearance of the true-8-to-9 error shown in Table 6.

References

Yoshua Bengio and Yves Grandvalet. No unbiased estimator of the variance of k-fold cross-validation. *Journal of Machine Learning Research*, 5:1089–1105, 2004. URL `https://www.jmlr.org/papers/volume5/grandvalet04a/grandvalet04a.pdf`. Full public PDF inspected; no DOI was verified in the cited record.

R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(2):179–188, 1936. doi: 10.1111/j.1469-1809.1936.tb02137.x.

Timothy Kassis, Vinayak Agarwal, Yuhuan He, Darshil Patel, and Aubrey M. Brueckner. Scientific agent skills: A library of procedural knowledge for research agents. arXiv preprint arXiv:2609.00065, 2026. URL `https://arxiv.org/abs/2609.00065`. Current arXiv record revised 2026-09-02; cited without a version suffix.

Olivier Ledoit and Michael Wolf. A well-conditioned estimator for large-dimensional covariance matrices. *Journal of Multivariate Analysis*, 88(2):365–411, 2004. doi: 10.1016/S0047-259X(03)00096-4.

Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Edouard Duchesnay. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(85):2825–2830, 2011. URL `https://jmlr.org/papers/v12/pedregosa11a.html`.

scikit-learn developers. load_digits: Load and return the digits dataset. scikit-learn 1.5 documentation, 2024a. URL `https://scikit-learn.org/1.5/modules/generated/sklearn.datasets.load_digits.html`. Dataset documentation page inspected 2026-09-08.

scikit-learn developers. discriminant_analysis.py. scikit-learn source tree, version 1.5.1, 2024b. URL `https://github.com/scikit-learn/scikit-learn/blob/1.5.1/sklearn/discriminant_analysis.py`. Functions _cov, _class_cov, and _solve_lstsq inspected; versioned source locator.

scikit-learn developers. _shrunk_covariance.py. scikit-learn source tree, version 1.5.1, 2024c. URL `https://github.com/scikit-learn/scikit-learn/blob/1.5.1/sklearn/covariance/_shrunk_covariance.py`. Functions shrunk_covariance and ledoit_wolf inspected; versioned source locator.