

calzone: A Python package for measuring calibration of probabilistic models for classification

Kwok Lung Fan¹, Gene Pennello¹, Qi Liu¹, Nicholas Petrick¹, Ravi K. Samala¹, Frank W. Samuelson¹, Yee Lam Elim Thompson¹, and Qian Cao¹

¹ U.S. Food and Drug Administration ¶ Corresponding author

DOI: [10.xxxxxx/draft](https://doi.org/10.xxxxxx/draft)

Software

- [Review](#)
- [Repository](#)
- [Archive](#)

Editor: [Open Journals](#)

Reviewers:

- [@openjournals](#)

Submitted: 01 January 1970

Published: unpublished

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](#)).

Summary

calzone is a Python package for evaluating the calibration of probabilistic outputs of classifier models. It provides a set of functions for visualizing calibration and computing of calibration metrics given a representative dataset with the model's predictions and the true class labels. The metrics provided in calzone include: Expected Calibration Error (ECE), Maximum Calibration Error (MCE), Hosmer-Lemeshow (HL) statistic, Integrated Calibration Index (ICI), Spiegelhalter's Z-statistics and Cox's calibration slope/intercept. The package is designed with versatility in mind. For many of the metrics, users can adjust the binning scheme and toggle between top-class or class-wise calculations.

Statement of need

Classification is one of the most common applications in machine learning. Examination of the discrimination performance (resolution), such as Area under the curve (AUC) or Sensitivity (Se), also known as true positive rate or Specificity (Sp, also known as 1 - false positive rate) are also used to evaluate model performance Hastie et al. (2001). These metrics may be sufficient if the output of the model is not meant to be a calibrated probability.

Diamond (1992) showed that the resolution (i.e., high performance) of a model does not indicate the reliability/calibration of the model. Calibration is the agreement between predicted and true probabilities, $P(D = 1|\hat{p} = p) = p$, defined as moderate calibration by Van Calster & Steyerberg (2018) and also known as model reliability. Bröcker (2009) later showed that any proper scoring rule can be decomposed into the resolution and reliability. Thus, a model with high resolution may still lack reliability. In high-risk applications like medical diagnosis, reliability aids interpretability for treatment decisions.

While existing libraries such as scikit-learn include basic tools like reliability diagrams and expected calibration error, they lack support for more comprehensive and flexible evaluation metrics—such as reliability diagrams with error bars, class-conditional calibration error, different binning schemes, or statistical significance testing for miscalibration. Other libraries, such as ml-calibration, uncertainty-toolbox, and pycalleva, primarily focus on only one aspect of calibration. For example, ml-calibration provides advanced controls for plotting reliability diagrams and computing smooth expected calibration error but does not include statistical tests for miscalibration (Błasiok & Nakkiran, 2024). The uncertainty-toolbox focuses on calibration methods rather than assessment (Chung et al., 2021). The pycalleva package overlaps with many functionalities in calzone, but it does not support Cox's calibration analysis, Wald intervals for reliability, or custom curve fitting methods for expected calibration error (Martin Weigl, 2022).

41 The calzone package offers functions and classes for visualizing and evaluating calibration
42 metrics using representative datasets.

43 In contrast, calzone emphasizes diagnostic tools for calibration assessment. It includes a wider
44 set of calibration metrics, statistical tests (e.g., hypothesis testing for miscalibration), and
45 visualization tools tailored for classification tasks with multiple classes. The package is designed
46 to help users not only visualize miscalibration but also quantify and statistically validate it in a
47 consistent and interpretable way.

48 Software description

49 Input data

50 To evaluate the calibration of a model, users need a representative dataset from the intended
51 population. The dataset should contain the true class labels and the model's predicted
52 probabilities. In calzone, the dataset can be a CSV file or two NumPy arrays containing true
53 labels and predicted probabilities.

54 Reliability Diagram

55 The reliability diagram is a graphical representation of the calibration([Bröcker & Smith, 2007](#);
56 [Murphy & Winkler, 1977](#)). It groups the predicted probabilities into bins and plots the mean
57 predicted probability against the empirical frequency in each bin. The reliability diagram can
58 be used to qualitatively assess the calibration of the model. The confidence intervals of the
59 empirical frequency are calculated using the Wilson's score interval ([Wilson, 1927](#)).

```
from calzone.utils import reliability_diagram
from calzone.vis import plot_reliability_diagram
reliability, confidence, bin_edges, bin_counts = reliability_diagram(
    labels,
    probs,
    num_bins=15,
    class_to_plot=1
)

plot_reliability_diagram(
    reliability,
    confidence,
    bin_counts,
    error_bar=True,
    title='Reliability diagram'
```

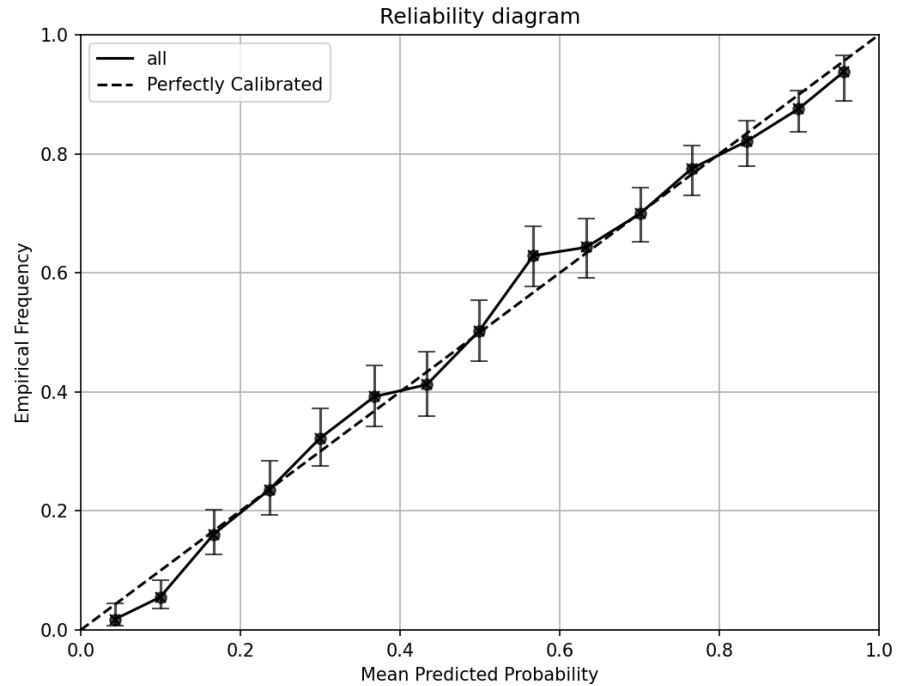


Figure 1: Reliability Diagram for class 1 with simulated data.

Calibration metrics

calzone provides functions to compute various calibration metrics, including methods to compute expected calibration error and statistical tests to assess calibration. These functions provide quantitative metrics for users to evaluate the calibration performance of the model. The `CalibrationMetrics()` class allows the user to compute the calibration metrics in a more convenient way. The following are metrics that are currently supported in calzone:

Expected Calibration Error (ECE) and Maximum Calibration Error (MCE)

Expected Calibration Error (ECE) and Maximum Calibration Error (MCE) (Guo et al., 2017; Pakdaman Naeini et al., 2015) measure the average and maximum deviation between predicted and true probabilities. calzone supports equal-width (ECE-H) and equal-count (ECE-C) binning. Users can compute these metrics for the top-class (highest probability) or class-of-interest (one-vs-rest classification).

Hosmer-Lemeshow statistic (HL)

The Hosmer-Lemeshow (HL) test (Hosmer & Lemeshow, 1980) evaluates model calibration using a chi-square test comparing observed and expected events in bins. The null hypothesis is that the model is well calibrated. calzone supports equal-width (ECE-H) and equal-count (ECE-C) binning. The test statistic is:

$$HL = \sum_{m=1}^M \frac{(O_{1,m} - E_{1,m})^2}{E_{1,m} \left(1 - \frac{E_{1,m}}{N_m}\right)} \sim \chi_{M-2}^2$$

where $E_{1,m}$ and $O_{1,m}$ are the expected and observed events in the m^{th} bin, N_m is the total observations in the bin, and M is the number of bins. For validation sets, the degrees of

freedom change from $M - 2$ to M (Hosmer Jr et al., 2013). The increase in degree of freedom for validation samples has often been overlooked but it is crucial for the test to maintain the correct type 1 error rate. In calzone, the default is $M - 2$, adjustable via the `df` parameter.

Cox's calibration slope/intercept

Cox's calibration slope/intercept assesses model calibration without binning (Cox, 1958). A logistic regression is fit with predicted odds ($\frac{p}{1-p}$) as the independent variable and the outcome as the dependent variable. Perfect calibration is indicated by a slope of 1 and intercept of 0. To test calibration, fit the intercept with slope fixed at 1; if the intercept differs from 0, the model is not calibrated. Similarly, fit the slope with intercept fixed at 0; if the slope differs from 1, the model is not calibrated. Alternatively, fit both simultaneously using a bivariate distribution (McCullagh & Nelder, 1989). This feature is not in calzone, but users can manually test using the covariance matrix.

A slope >1 indicates overconfidence at high probabilities and underconfidence at low probabilities, while a slope <1 indicates the opposite. A positive intercept indicates general overconfidence. Even with ideal slope and intercept, non-linear miscalibration may still exist.

Integrated calibration index (ICI)

The Integrated Calibration Index (ICI) measures the average deviation between predicted and true probabilities using curve smoothing techniques (Austin & Steyerberg, 2019). It is calculated as:

$$ICI = \frac{1}{n} \sum_{i=1}^n |f(p_i) - p_i|$$

where f is the fitting function and p is the predicted probability. Typically, Locally Weighted Scatterplot Smoothing (LOWESS) is used, but any curve fitting method can be applied. calzone supports both Cox ICI and LOWESS ICI, allowing users to choose their preferred method. Users should visualize the fitting results to avoid overfitting or underfitting, as flexible methods like LOWESS are sensitive to span and delta parameters.

Spiegelhalter's Z-test

Spiegelhalter's Z-test is a test of calibration proposed by Spiegelhalter in 1986 (Spiegelhalter, 1986). It uses the fact that the Brier score can be decomposed into:

$$B = \frac{1}{N} \sum_{i=1}^N (x_i - p_i)^2 = \frac{1}{N} \sum_{i=1}^N (x_i - p_i)(1 - 2p_i) + \frac{1}{N} \sum_{i=1}^N p_i(1 - p_i)$$

And the test statistic (TS) of Z test is defined as:

$$Z = \frac{B - E(B)}{\sqrt{\text{Var}(B)}} = \frac{\sum_{i=1}^N (x_i - p_i)(1 - 2p_i)}{\sum_{i=1}^N (1 - 2p_i)^2 p_i(1 - p_i)}$$

and it is asymptotically distributed as a standard normal distribution.

Metrics class

calzone also provides a class called `CalibrationMetrics()` to calculate all the metrics mentioned above. The function will return a dictionary containing the metrics name and their values. The metrics can be specified as a list of strings. The string 'all' can be used to calculate all the metrics.

```
from calzone.metrics import CalibrationMetrics
```

```
metrics = CalibrationMetrics(class_to_calculate=1)

metrics.calculate_metrics(
    labels,
    probs,
    metrics='all'
)
```

Other features

Confidence intervals

calzone also provides functionality to compute confidence intervals for all metrics using bootstrapping. The user can specify the number of bootstrap samples and the confidence level.

```
from calzone.metrics import CalibrationMetrics

metrics = CalibrationMetrics(class_to_calculate=1)

CalibrationMetrics.bootstrap(
    labels,
    probs,
    metrics='all',
    n_samples=1000
)
```

and a structured NumPy array will be returned.

Subgroup analysis

calzone will perform subgroup analysis by default in the command line user interface. If the user input CSV file contains a subgroup column, the program will compute metrics for the entire dataset and for each subgroup. A detailed description of the input format can be found in the documentation.

Prevalence adjustment

calzone offers prevalence adjustment to correct for differences in disease prevalence between training and testing data. Calibration is based on posterior probability, so a shift in prevalence can cause miscalibration. The adjusted probability is calculated as:

$$P'(D = 1|\hat{p} = p) = \frac{\eta'/(1 - \eta')}{(1/p - 1)(\eta/(1 - \eta))} = p'$$

where η is the testing data prevalence, η' is the training data prevalence, and p is the predicted probability. The optimal η' is found by minimizing cross-entropy loss, or users can specify η' directly if known (Chen et al., 2018; Gu & Pepe, 2010; Horsch et al., 2008; Tian et al., 2020).

Multiclass extension

calzone supports multiclass classification using a 1-vs-rest approach or top-class calibration. In top-class calibration, class 1 probability is the highest predicted probability, and class 0 is 1 minus this probability. Metrics interpretation may change in this transformation.

135 Verification of methods

136 calzone results were compared with external packages for accuracy. Reliability diagrams were
137 verified with `sklearn.calibration.calibration_curve()` (Pedregosa et al., 2011), top-class
138 ECE and Spiegelhalter's Z scores with `MAPIE` (Taquet et al., 2022), and Hosmer-Lemeshow
139 statistic with `ResourceSelection` (Lele et al., 2024) in R. Differences were within 0.1%,
140 confirming consistency. Verification codes are in the documentation.

141 Command line interface

142 calzone offers a command line interface for visualizing calibration curves, calculating metrics,
143 and confidence intervals. Run `python cal_metrics.py -h` for help.

144 Acknowledgements

145 The mention of commercial products, their sources, or their use in connection with material
146 reported herein is not to be construed as either an actual or implied endorsement of such
147 products by the Department of Health and Human Services. This is a contribution of the U.S.
148 Food and Drug Administration and is not subject to copyright.

149 The authors acknowledge the Research Participation Program at the Center for Devices and
150 Radiological Health administered by the Oak Ridge Institute for Science and Education through
151 an interagency agreement between the U.S. Department of Energy and the U.S. Food and
152 Drug Administration.

153 Conflicts of interest

154 The authors declare no conflicts of interest.

155 References

- 156 Austin, P. C., & Steyerberg, E. W. (2019). The integrated calibration index (ICI) and related
157 metrics for quantifying the calibration of logistic regression models. *Statistics in Medicine*,
158 38(21), 4051–4065. <https://doi.org/10.1002/sim.8281>
- 159 Błasiok, J., & Nakkiran, P. (2024). Smooth ECE: Principled reliability diagrams via kernel
160 smoothing. *The Twelfth International Conference on Learning Representations*. <https://openreview.net/forum?id=XwiA1nDahv>
- 162 Bröcker, J. (2009). Reliability, sufficiency, and the decomposition of proper scores. *Quarterly*
163 *Journal of the Royal Meteorological Society*, 135(643), 1512–1519. <https://doi.org/10.1002/qj.456>
- 165 Bröcker, J., & Smith, L. A. (2007). Increasing the reliability of reliability diagrams. *Weather*
166 *and Forecasting*, 22(3), 651–661. <https://doi.org/10.1175/WAF993.1>
- 167 Chen, W., Sahiner, B., Samuelson, F., Pezeshk, A., & Petrick, N. (2018). Calibration of
168 medical diagnostic classifier scores to the probability of disease. *Statistical Methods in*
169 *Medical Research*, 27(5), 1394–1409. <https://doi.org/10.1177/0962280216661371>
- 170 Chung, Y., Char, I., Guo, H., Schneider, J., & Neiswanger, W. (2021). Uncertainty toolbox:
171 An open-source library for assessing, visualizing, and improving uncertainty quantification.
172 *arXiv Preprint arXiv:2109.10254*.
- 173 Cox, D. R. (1958). Two further applications of a model for binary regression. *Biometrika*,
174 45(3-4), 562–565. <https://doi.org/10.1093/biomet/45.3-4.562>

- 175 Diamond, G. A. (1992). What price perfection? Calibration and discrimination of clinical
176 prediction models. *Journal of Clinical Epidemiology*, 45(1), 85–89. [https://doi.org/10.1016/0895-4356\(92\)90192-P](https://doi.org/10.1016/0895-4356(92)90192-P)
177
- 178 Gu, W., & Pepe, M. S. (2010). Estimating the diagnostic likelihood ratio of a continuous
179 marker. *Biostatistics*, 12(1), 87–101. <https://doi.org/10.1093/biostatistics/kxq045>
- 180 Guo, C., Pleiss, G., Sun, Y., & Weinberger, K. Q. (2017). On calibration of modern neural
181 networks. In D. Precup & Y. W. Teh (Eds.), *Proceedings of the 34th international
182 conference on machine learning* (Vol. 70, pp. 1321–1330). PMLR. <https://proceedings.mlr.press/v70/guo17a.html>
183
- 184 Hastie, T., Tibshirani, R., & Friedman, J. (2001). *The elements of statistical learning*. Springer
185 New York Inc.
- 186 Horsch, K., Giger, M. L., & Metz, C. E. (2008). Prevalence scaling: Applications to an
187 intelligent workstation for the diagnosis of breast cancer. *Academic Radiology*, 15(11),
188 1446–1457. <https://doi.org/10.1016/j.acra.2008.04.022>
- 189 Hosmer, D. W., & Lemeshow, S. (1980). Goodness of fit tests for the multiple logistic
190 regression model. *Communications in Statistics - Theory and Methods*, 9(10), 1043–1069.
191 <https://doi.org/10.1080/03610928008827941>
- 192 Hosmer Jr, D. W., Lemeshow, S., & Sturdivant, R. X. (2013). *Applied logistic regression*.
193 John Wiley & Sons.
- 194 Lele, S. R., Keim, J. L., & Solymos, P. (2024). *ResourceSelection: Resource selection
195 (probability) functions for use-availability data*. <https://doi.org/10.32614/cran.package.resourceselection>
196
- 197 Martin Weigl, M. A. S. (2022). *Pycaleva*. <https://github.com/MartinWeigl/pycaleva>.
- 198 McCullagh, P., & Nelder, J. A. (1989). *Generalized linear models*. Chapman & Hall / CRC.
199 <https://doi.org/10.1201/9781439891148-8>
- 200 Murphy, A. H., & Winkler, R. L. (1977). Reliability of subjective probability forecasts of
201 precipitation and temperature. *Journal of the Royal Statistical Society. Series C (Applied
202 Statistics)*, 26(1), 41–47. <https://doi.org/10.2307/2346866>
- 203 Pakdaman Naeini, M., Cooper, G., & Hauskrecht, M. (2015). Obtaining well calibrated
204 probabilities using bayesian binning. *Proceedings of the AAAI Conference on Artificial
205 Intelligence*, 29(1). <https://doi.org/10.1609/aaai.v29i1.9602>
- 206 Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M.,
207 Prettenhofer, P., Weiss, R., Dubourg, V., & others. (2011). Scikit-learn: Machine learning
208 in python. *Journal of Machine Learning Research*, 12(Oct), 2825–2830.
- 209 Spiegelhalter, D. J. (1986). Probabilistic prediction in patient management and clinical trials.
210 *Statistics in Medicine*, 5(5), 421–433. <https://doi.org/10.1002/sim.4780050506>
- 211 Taquet, V., Blot, V., Morzadec, T., Lacombe, L., & Brunel, N. (2022). MAPIE: An open-source
212 library for distribution-free uncertainty quantification. *arXiv Preprint arXiv:2207.12274*.
- 213 Tian, J., Liu, Y.-C., Glaser, N., Hsu, Y.-C., & Kira, Z. (2020). Posterior re-calibration
214 for imbalanced datasets. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan,
215 & H. Lin (Eds.), *Advances in neural information processing systems* (Vol. 33, pp.
216 8101–8113). Curran Associates, Inc. [https://proceedings.neurips.cc/paper_files/paper/
217 2020/file/5ca359ab1e9e3b9c478459944a2d9ca5-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2020/file/5ca359ab1e9e3b9c478459944a2d9ca5-Paper.pdf)
- 218 Van Calster, B., & Steyerberg, E. W. (2018). Calibration of prognostic risk scores. In
219 *Wiley StatsRef: Statistics reference online* (pp. 1–10). John Wiley & Sons, Ltd. <https://doi.org/10.1002/9781118445112.stat08078>
220

- 221 Wilson, E. B. (1927). Probable inference, the law of succession, and statistical inference.
222 *Journal of the American Statistical Association*, 22(158), 209–212. [https://doi.org/10.](https://doi.org/10.2307/2276774)
223 [2307/2276774](https://doi.org/10.2307/2276774)

DRAFT