

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Published: unpublished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. Metrics associated with discrimination performance (resolution), such as Area under the curve (AUC), Sensitivity (Se, true positive rate), and Specificity (Sp, 1 - false positive rate) are typically used to characterize classification performance Hastie et al. (2001). These metrics may be sufficient if the outputs of the model are not meant to be interpreted as a probability.

However, 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), 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 medical applications such as computer-aided diagnosis, reliability enables the correct interpretation of model output, and for making downstream 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. This is also the case with other calibration-focused libraries, such as ml-calibration, uncertainty-toolbox, and pycalleva. 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 (Blasiok & 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). In contrast, Calzone emphasizes the evaluation of calibration. It

42 features a comprehensive set of calibration metrics, statistical tests (e.g., hypothesis testing
43 for miscalibration), and visualization tools tailored for many types of classification tasks (e.g.,
44 multi-class metrics). The package is designed to help users not only visualize miscalibration
45 but also quantify and statistically validate it in a consistent and interpretable way.

46 Software description

47 Input data

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

52 Reliability Diagram

53 The reliability diagram is a graphical representation of the calibration (Bröcker & Smith, 2007;
54 Murphy & Winkler, 1977). It groups the predicted probabilities into bins and plots the mean
55 predicted probability against the empirical frequency in each bin. The reliability diagram can
56 be used to qualitatively assess the calibration of the model. The confidence intervals of the
57 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'
)
```

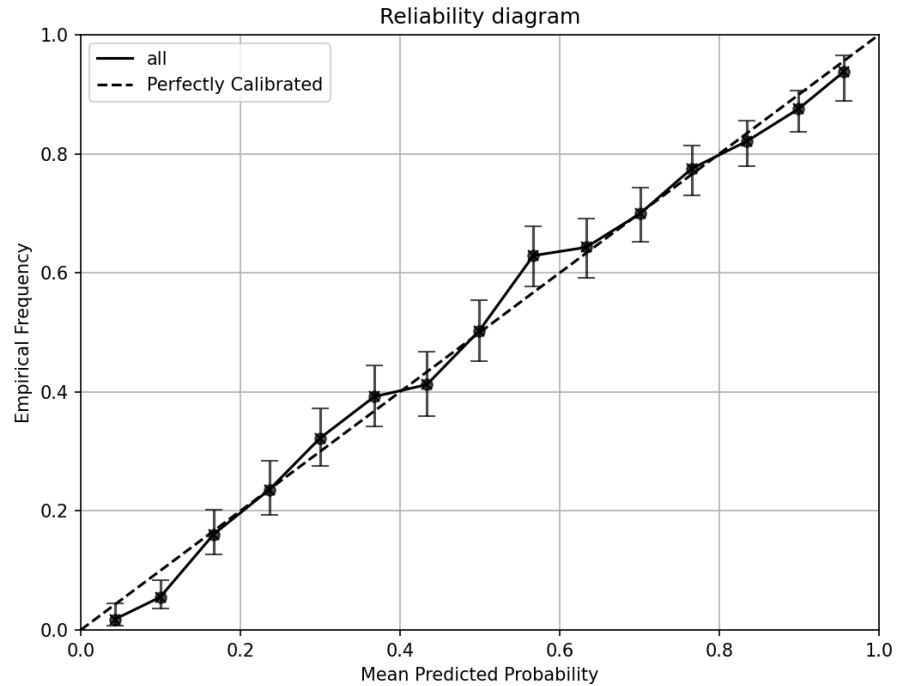


Figure 1: Reliability Diagram for class 1 with simulated data.

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.

133 Verification of methods

134 To ensure the accuracy and reliability of the metrics implemented in calzone, we performed
135 comprehensive validation against established external packages. Reliability diagrams were
136 compared with `sklearn.calibration.calibration_curve()` (Pedregosa et al., 2011), top-
137 class ECE and Spiegelhalter's Z scores were validated against MAPE (Taquet et al., 2022), and
138 the Hosmer-Lemeshow statistic was checked against ResourceSelection (Lele et al., 2024) in
139 R. Additional tests were conducted using the `relplot` and `pycalleva` Python packages to further
140 confirm metric consistency. All differences were within 0.1%, demonstrating strong agreement.
141 These validation tests are documented in `test_results.py`. Furthermore, synthetic data tests
142 (see `test_metrics.py`) were used to confirm the expected behavior of the calibration metrics
143 under controlled conditions.

144 Command line interface

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

147 Acknowledgements

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

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

156 Conflicts of interest

157 The authors declare no conflicts of interest.

158 References

- 159 Austin, P. C., & Steyerberg, E. W. (2019). The integrated calibration index (ICI) and related
160 metrics for quantifying the calibration of logistic regression models. *Statistics in Medicine*,
161 38(21), 4051–4065. <https://doi.org/10.1002/sim.8281>
- 162 Blasiok, J., & Nakkiran, P. (2024). Smooth ECE: Principled reliability diagrams via kernel
163 smoothing. *The Twelfth International Conference on Learning Representations, ICLR 2024*,
164 Vienna, Austria, May 7-11, 2024. <https://openreview.net/forum?id=XwiA1nDahv>
- 165 Bröcker, J. (2009). Reliability, sufficiency, and the decomposition of proper scores. *Quarterly*
166 *Journal of the Royal Meteorological Society*, 135(643), 1512–1519. <https://doi.org/10.1002/qj.456>
- 167 Bröcker, J., & Smith, L. A. (2007). Increasing the reliability of reliability diagrams. *Weather*
168 *and Forecasting*, 22(3), 651–661. <https://doi.org/10.1175/WAF993.1>
- 170 Chen, W., Sahiner, B., Samuelson, F., Pezeshk, A., & Petrick, N. (2018). Calibration of
171 medical diagnostic classifier scores to the probability of disease. *Statistical Methods in*
172 *Medical Research*, 27(5), 1394–1409. <https://doi.org/10.1177/0962280216661371>

- 173 Chung, Y., Char, I., Guo, H., Schneider, J., & Neiswanger, W. (2021). Uncertainty toolbox:
174 An open-source library for assessing, visualizing, and improving uncertainty quantification.
175 *arXiv Preprint arXiv:2109.10254*. <https://doi.org/10.48550/arXiv.2109.10254>
- 176 Cox, D. R. (1958). Two further applications of a model for binary regression. *Biometrika*,
177 45(3-4), 562–565. <https://doi.org/10.1093/biomet/45.3-4.562>
- 178 Diamond, G. A. (1992). What price perfection? Calibration and discrimination of clinical
179 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)
- 180
- 181 Gu, W., & Pepe, M. S. (2010). Estimating the diagnostic likelihood ratio of a continuous
182 marker. *Biostatistics*, 12(1), 87–101. <https://doi.org/10.1093/biostatistics/kxq045>
- 183 Guo, C., Pleiss, G., Sun, Y., & Weinberger, K. Q. (2017). On calibration of modern neural
184 networks. In D. Precup & Y. W. Teh (Eds.), *Proceedings of the 34th international*
185 *conference on machine learning* (Vol. 70, pp. 1321–1330). PMLR. <https://proceedings.mlr.press/v70/guo17a.html>
- 186
- 187 Hastie, T., Tibshirani, R., & Friedman, J. (2001). *The elements of statistical learning*. Springer
188 New York Inc. ISBN: 978-0387848570
- 189 Horsch, K., Giger, M. L., & Metz, C. E. (2008). Prevalence scaling: Applications to an
190 intelligent workstation for the diagnosis of breast cancer. *Academic Radiology*, 15(11),
191 1446–1457. <https://doi.org/10.1016/j.acra.2008.04.022>
- 192 Hosmer, D. W., & Lemeshow, S. (1980). Goodness of fit tests for the multiple logistic
193 regression model. *Communications in Statistics - Theory and Methods*, 9(10), 1043–1069.
194 <https://doi.org/10.1080/03610928008827941>
- 195 Hosmer Jr, D. W., Lemeshow, S., & Sturdivant, R. X. (2013). *Applied logistic regression*.
196 John Wiley & Sons. ISBN: 9781118548387
- 197 Lele, S. R., Keim, J. L., & Solymos, P. (2024). *ResourceSelection: Resource selection*
198 *(probability) functions for use-availability data*. <https://doi.org/10.32614/cran.package.resourceselection>
- 199
- 200 Martin Weigl, M. A. S. (2022). *Pycaleva*. <https://github.com/MartinWeigl/pycaleva>.
- 201 McCullagh, P., & Nelder, J. A. (1989). *Generalized linear models*. Chapman & Hall / CRC.
202 <https://doi.org/10.1201/9781439891148-8>
- 203 Murphy, A. H., & Winkler, R. L. (1977). Reliability of subjective probability forecasts of
204 precipitation and temperature. *Journal of the Royal Statistical Society. Series C (Applied*
205 *Statistics)*, 26(1), 41–47. <https://doi.org/10.2307/2346866>
- 206 Pakdaman Naeini, M., Cooper, G., & Hauskrecht, M. (2015). Obtaining well calibrated
207 probabilities using bayesian binning. *Proceedings of the AAAI Conference on Artificial*
208 *Intelligence*, 29(1). <https://doi.org/10.1609/aaai.v29i1.9602>
- 209 Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M.,
210 Prettenhofer, P., Weiss, R., Dubourg, V., & others. (2011). Scikit-learn: Machine learning
211 in python. *Journal of Machine Learning Research*, 12(Oct), 2825–2830.
- 212 Spiegelhalter, D. J. (1986). Probabilistic prediction in patient management and clinical trials.
213 *Statistics in Medicine*, 5(5), 421–433. <https://doi.org/10.1002/sim.4780050506>
- 214 Taquet, V., Blot, V., Morzadec, T., Lacombe, L., & Brunel, N. (2022). MAPIE: An open-source
215 library for distribution-free uncertainty quantification. *arXiv Preprint arXiv:2207.12274*.
216 <https://doi.org/10.48550/arXiv.2207.12274>
- 217 Tian, J., Liu, Y.-C., Glaser, N., Hsu, Y.-C., & Kira, Z. (2020). Posterior re-calibration
218 for imbalanced datasets. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan,

- 219 & H. Lin (Eds.), *Advances in neural information processing systems* (Vol. 33, pp.
220 8101–8113). Curran Associates, Inc. [https://proceedings.neurips.cc/paper_files/paper/](https://proceedings.neurips.cc/paper_files/paper/2020/file/5ca359ab1e9e3b9c478459944a2d9ca5-Paper.pdf)
221 [2020/file/5ca359ab1e9e3b9c478459944a2d9ca5-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2020/file/5ca359ab1e9e3b9c478459944a2d9ca5-Paper.pdf)
- 222 Van Calster, B., & Steyerberg, E. W. (2018). Calibration of prognostic risk scores. In
223 *Wiley StatsRef: Statistics reference online* (pp. 1–10). John Wiley & Sons, Ltd. [https:](https://doi.org/10.1002/9781118445112.stat08078)
224 [//doi.org/10.1002/9781118445112.stat08078](https://doi.org/10.1002/9781118445112.stat08078)
- 225 Wilson, E. B. (1927). Probable inference, the law of succession, and statistical inference.
226 *Journal of the American Statistical Association*, 22(158), 209–212. [https://doi.org/10.](https://doi.org/10.2307/2276774)
227 [2307/2276774](https://doi.org/10.2307/2276774)

DRAFT