
plot_{metric}
Release 0.0.4

Aug 22, 2022

Contents:

1	Indices and tables	9
	Index	11


```
class plot_metric.functions.BinaryClassification(y_true, y_pred, labels, thresh-
old=0.5, seaborn_style='darkgrid',
matplotlib_style=None)
```

Initialize class.

Parameters

- **y_true** (array, list, shape = [n_sample]) – True binary labels.
- **y_pred** (array, list, shape = [n_sample]) – Target scores, can either be probability estimates of the positive class, confidence values, or non-thresholded measure of decisions (as returned by “decision_function” on some classifiers).
- **labels** (array, list, shape = [n_class]) – String or int of to define targeted classes.
- **threshold** (float [0-1], default=0.5,) – Classification threshold (or decision threshold). More information about threshold : - <https://developers.google.com/machine-learning/crash-course/classification/thresholding> - https://en.wikipedia.org/wiki/Threshold_model
- **seaborn_style** (string, default='darkgrid') – Set the style of seaborn library, preset available with seaborn : darkgrid, whitegrid, dark, white, and ticks. See <https://seaborn.pydata.org/tutorial/aesthetics.html#seaborn-figure-styles> for more info.
- **matplotlib_style** (string, default=None) – Set the style of matplotlib. Find all preset here : https://matplotlib.org/3.1.0/gallery/style_sheets/style_sheets_reference.html Or with the following code :
- **code**: (.) – python: import matplotlib.style as style style.available

```
get_function_parameters (function, as_df=False)
```

Function to get all available parameters for a given function.

Parameters

- **function** (func) – Function parameter’s wanted.
- **as_df** (boolean, default=False) – Set to True to return a dataframe with parameters instead of dictionary.

Returns param_dict – Dictionary containing parameters for the given function and their default value.

Return type dict

```
plot_class_distribution (threshold=None, display_prediction=True, alpha=0.5, jitter=0.3, pal_colors=None, display_violin=True, c_violin='white',
strip_marker_size=4, strip_lw_edge=None, strip_c_edge=None,
ls_thresh_line=':', c_thresh_line='red', lw_thresh_line=2, title=None)
```

Plot distribution of the predictions for each classes.

Note : Threshold here is importante because it define colors for True Positive, False Negative, True Negative and False Positive.

Parameters

- **threshold** (float, default=0.5) – Threshold to determinine the rate between positive and negative values of the classification.
- **display_prediction** (bool, default=True) – Display the point representing each predictions.

- **alpha** (*float*, *default=0.5*) – Transparency of each predicted point.
- **jitter** (*float*, *default=0.3*) – Amount of jitter (only along the categorical axis) to apply. This can be useful when you have many points and they overlap, so that it is easier to see the distribution. You can specify the amount of jitter (half the width of the uniform random variable support), or just use True for a good default. See : <https://seaborn.pydata.org/generated/seaborn.stripplot.html>
- **pal_colors** (*palette name, list, or dict, optional, default=["#00C853", "#FF8A80", "#C5E1A5", "#D50000"]*) – Colors to use for the different levels of the hue variable. Should be something that can be interpreted by `color_palette()`, or a dictionary mapping hue levels to matplotlib colors. See : <https://seaborn.pydata.org/generated/seaborn.stripplot.html>
- **display_violin** (*bool*, *default=True*) – Display violin plot.
- **c_violin** (*string*, *default='white'*) – Color of the violinplot.
- **strip_marker_size** (*int*, *default='4'*) – Size of marker representing predictions.
- **strip_lw_edge** (*float*, *default=None*) – Size of the linewidth for the edge of point prediction.
- **strip_c_edge** (*string*, *default=None*) – Color of the linewidth for the edge of point prediction.
- **ls_thresh_line** (*string*, *default=':'*) – Linestyle for the threshold line.
- **c_thresh_line** (*string*, *default='red'*) – Color for the threshold line.
- **lw_thresh_line** (*float*, *default=2*) – Line width of the threshold line.
- **title** (*string*, *default=None*) – String for the title of the graphic.

Returns

- *DataFrame with the following column*
- *- True Class*
- *- Predicted Proba*
- *- Predicted Type*
- *- Predicted Class*

plot_confusion_matrix (*threshold=None*, *normalize=False*, *title='Confusion matrix'*, *cmap=<matplotlib.colors.LinearSegmentedColormap object>*, *colorbar=True*, *label_rotation=45*)

Plots the confusion matrix.

Parameters

- **threshold** (*float*, *default=0.5*) – Threshold to determine the rate between positive and negative values of the classification.
- **normalize** (*bool*, *default=False*) – Set to True to normalize matrix and make matrix coefficient between 0 and 1.
- **title** (*string*, *default="Confusion matrix",*) – Set title of the plot.
- **cmap** (*colormap*, *default=plt.cm.Reds*) – Colormap of the matrix. See https://matplotlib.org/examples/color/colormaps_reference.html to find all available colormap.
- **colorbar** (*bool*, *default=True*) – Display color bar beside matrix.

- **label_rotation** (*int*, *default=45*) – Degree of rotation for x_axis labels.

Returns **cm** – Return confusion_matrix computed by sklearn.metrics.confusion_matrix

Return type array, shape=[n_classes, n_classes]

plot_precision_recall_curve (*threshold=None*, *plot_threshold=True*, *beta=1*, *linewidth=2*, *fscore_iso=[0.2, 0.4, 0.6, 0.8]*, *iso_alpha=0.7*, *y_text_margin=0.03*, *x_text_margin=0.2*, *c_pr_curve='black'*, *c_mean_prec='red'*, *c_thresh_lines='black'*, *c_f1_iso='grey'*, *c_thresh_point='red'*, *ls_pr_curve='-'*, *ls_mean_prec='-'*, *ls_thresh=':'*, *ls_fscore_iso=':'*, *marker_pr_curve=None*, *title='Precision and Recall Curve'*)

Compute and plot the precision-recall curve.

Note : this implementation is restricted to binary classification only. See MultiClassClassification for multi-classes implementation.

F1-iso are curve where a given f1-score is constant.

We also consider the use of F_beta-score, change the parameter beta to use an other f-score. “Two other commonly used F measures are the F₂ measure, which weighs recall higher than precision (by placing more emphasis on false negatives), and the F_{0.5} measure, which weighs recall lower than precision (by attenuating the influence of false negatives). (Wiki)”

Parameters

- **threshold** (*float*, *default=0.5*) – Threshold to determine the rate between positive and negative values of the classification.
- **plot_threshold** (*boolean*, *default=True*) – Plot or not precision and recall lines for the given threshold.
- **beta** (*float*, *default=1*,) – Set beta to another float to use a different f_beta score. See definition of f_beta-score for more information : https://en.wikipedia.org/wiki/F1_score
- **linewidth** (*float*, *default=2*) –
- **fscore_iso** (*array, list*, *default=[0.2, 0.4, 0.6, 0.8]*) – List of float f1-score. Set to None or empty list to remove plotting of iso.
- **iso_alpha** (*float*, *default=0.7*) – Transparency of iso-f1.
- **y_text_margin** (*float*, *default=0.03*) – Margin (y) of text threshold.
- **x_text_margin** (*float*, *default=0.2*) – Margin (x) of text threshold.
- **c_pr_curve** (*string*, *default='black'*) – Define the color of precision-recall curve.
- **c_mean_prec** (*string*, *default='red'*) – Define the color of mean precision line.
- **c_thresh** (*string*, *default='black'*) – Define the color of threshold lines.
- **c_f1_iso** (*string*, *default='grey'*) – Define the color of iso-f1 curve.
- **c_thresh_point** (*string*, *default='red'*) – Define the color of threshold point.
- **ls_pr_curve** (*string*, *default='-'*) – Define the linestyle of precision-recall curve.

- **ls_mean_prec** (*string*, *default*='--') – Define the linestyle of mean precision line.
- **ls_thresh** (*string*, *default*=':') – Define the linestyle of threshold lines.
- **ls_fscore_iso** (*string*, *default*=':') – Define the linestyle of iso-f1 curve.
- **marker_pr_curve** (*string*, *default*=None) – Define the marker of precision-recall curve.
- **title** (*string*, *default*="Precision and Recall Curve") – Set title of the figure.

Returns

- **prec** (*array*, *shape* = [*n_thresholds* + 1]) – Precision values such that element *i* is the precision of predictions with score \geq thresholds[*i*] and the last element is 1.
- **recall** (*array*, *shape* = [*n_thresholds* + 1]) – Decreasing recall values such that element *i* is the recall of predictions with score \geq thresholds[*i*] and the last element is 0.
- **thresh** (*array*, *shape* = [*n_thresholds* \leq len(np.unique(*y_pred*))]) – Increasing thresholds on the decision function used to compute precision and recall.

plot_roc_curve (*threshold*=None, *plot_threshold*=True, *linewidth*=2, *y_text_margin*=0.05, *x_text_margin*=0.2, *c_roc_curve*='black', *c_random_guess*='red', *c_thresh_lines*='black', *ls_roc_curve*='-', *ls_thresh_lines*=':', *ls_random_guess*='-', *title*='Receiver Operating Characteristic', *loc_legend*='lower right')

Compute and plot the ROC (Receiver Operating Characteristics) curve but also AUC (Area Under The Curve).

Note : for more information about ROC curve and AUC look at the reference given.

Moreover, this implementation is restricted to binary classification only. See [MultiClassClassification](#) for multi-classes implementation.

Parameters

- **threshold** (*float*, *default*=0.5) –
- **plot_threshold** (*boolean*, *default*=True) – Plot or not ROC lines for the given threshold.
- **linewidth** (*float*, *default*=2) –
- **y_text_margin** (*float*, *default*=0.03) – Margin (y) of text threshold.
- **x_text_margin** (*float*, *default*=0.2) – Margin (x) of text threshold.
- **c_roc_curve** (*string*, *default*='black') – Define the color of ROC curve.
- **c_random_guess** (*string*, *default*='red') – Define the color of random guess line.
- **c_thresh_lines** (*string*, *default*='black') – Define the color of threshold lines.
- **ls_roc_curve** (*string*, *default*='-') – Define the linestyle of ROC curve.
- **ls_thresh_lines** (*string*, *default*=':') – Define the linestyle of threshold lines.
- **ls_random_guess** (*string*, *default*='--') – Define the linestyle of random guess line.

- **title** (*string*, *default*='Receiver Operating Characteristic') – Set title of the figure.
- **loc_legend** (*string*, *default*='loc_legend') – Localisation of legend. Available string are the following :

Location String	Location Code
'best'	0
'upper right'	1
'upper left'	2
'lower left'	3
'lower right'	4
'right'	5
'center left'	6
'center right'	7
'lower center'	8
'upper center'	9
'center'	10

Returns

- **fpr** (*array*, *shape* = [>2]) – Increasing false positive rates such that element *i* is the false positive rate of predictions with score \geq thresholds[*i*].
- **tpr** (*array*, *shape* = [>2]) – Increasing true positive rates such that element *i* is the true positive rate of predictions with score \geq thresholds[*i*].
- **thresh** (*array*, *shape* = [*n_thresholds*]) – Decreasing thresholds on the decision function used to compute fpr and tpr. *thresholds*[0] represents no instances being predicted and is arbitrarily set to $\max(y_score) + 1$.
- **auc** (*float*)

References

plot_threshold (*threshold*=None, *beta*=1, *title*=None, *annotation*=True, *bbox_dict*=None, *bbox*=True, *arrow_dict*=None, *arrow*=True, *plot_fscore*=True, *plot_recall*=True, *plot_prec*=True, *plot_fscore_max*=True, *c_recall_line*='green', *lw_recall_line*=2, *ls_recall_line*='-', *label_recall*='Recall', *marker_recall*="", *c_prec_line*='blue', *lw_prec_line*=2, *ls_prec_line*='-', *label_prec*='Precision', *marker_prec*="", *c_fscr_line*='red', *lw_fscr_line*=2, *ls_fscr_line*='-', *label_fscr*=None, *marker_fscr*="", *marker_fscore_max*='o', *c_fscore_max*='red', *markersize_fscore_max*=5, *plot_threshold*=True, *c_thresh_line*='black', *lw_thresh_line*=2, *ls_thresh_line*='-', *plot_best_threshold*=True, *c_bestthresh_line*='black', *lw_bestthresh_line*=1, *ls_bestthresh_line*=':')

Plot precision - threshold, recall - threshold and fbeta-score - threshold curves. Also plot threshold line for a given threshold and threshold line for the best ratio between precision and recall.

Parameters

- **threshold** (*float*, *default*=0.5) – Threshold to determine the rate between positive and negative values of the classification.
- **beta** (*float*, *default*=1,) – Set beta to another float to use a different f_beta score. See definition of f_beta-score for more information : https://en.wikipedia.org/wiki/F1_score

- **title** (*string*, *default=None*) – String for the title of the graphic.
- **annotation** (*bool*, *default=True*) – Boolean to display annotation box with threshold, precision and recall score.
- **bbox_dict** (*dict*, *default*={'facecolor': 'none', } –
 'edgecolor': 'black', 'boxstyle': 'round', 'alpha': 0.4, 'pad': 0.3}
Set the parameters of the bbox annotation. See matplotlib [documentation](#) for more information.
- **bbox** (*bool*, *default=True*) – Boolean to display the bbox around annotation.
- **arrow_dict** (*dict*, *default*={'arrowstyle': "->", 'color': 'black'}) – Set the parameters of the bbox annotation. See matplotlib [documentation](#) for more information.
- **arrow** (*bool*, *default=True*) – Boolean to display the array for the annotation.
- **plot_fscore** (*bool*, *default=True*) – Boolean to plot the FBeta-Score curve.
- **plot_recall** (*bool*, *default=True*) – Boolean to plot the recall curve.
- **plot_prec** (*bool*, *default=True*) – Boolean to plot the precision curve.
- **plot_fscore_max** (*bool*, *default=True*) – Boolean to plot the point showing fbeta-score max.
- **c_recall_line** (*string*, *default*='green') – Color of the recall curve.
- **lw_recall_line** (*float*, *default*=2) – Linewidth of the recall curve.
- **ls_recall_line** (*string*, *default*='-') – Linestyle of the recall curve.
- **label_recall** (*string*, *default*='Recall') – Label of the recall curve.
- **marker_recall** (*string*, *default*='') – Marker of the recall curve.
- **c_prec_line** (*string*, *default*='green') – Color of the prec curve.
- **lw_prec_line** (*float*, *default*=2) – Linewidth of the prec curve.
- **ls_prec_line** (*string*, *default*='-') – Linestyle of the prec curve.
- **label_prec** (*string*, *default*='prec') – Label of the prec curve.
- **marker_prec** (*string*, *default*='') – Marker of the prec curve.
- **c_fscr_line** (*string*, *default*='green') – Color of the fscr curve.
- **lw_fscr_line** (*float*, *default*=2) – Linewidth of the fscr curve.
- **ls_fscr_line** (*string*, *default*='-') – Linestyle of the fscr curve.
- **label_fscr** (*string*, *default*='fscr') – Label of the fscr curve.
- **marker_fscr** (*string*, *default*='') – Marker of the fscr curve.
- **marker_fscore_max** (*string*, *default*='o') – Marker for the fscore max point.
- **c_fscore_max** (*string*, *default*='red') – Color for the fscore max point.
- **markersize_fscore_max** (*float*, *default*=5) – Marker size for the fscore max point.
- **plot_threshold** (*bool*, *default=True*) – Plot a line at the given threshold.

- **c_thresh_line**(*string*, *default*='black') – Color for the threshold line.
- **lw_thresh_line**(*float*, *default*=2) – Linewidth for the threshold line.
- **ls_thresh_line**(*string*, *default*='--') – Linestyle for the threshold line.
- **plot_best_threshold**(*bool*, *default*=True) – Plot a line at the best threshold (best ratio precision-recall).
- **c_bestthresh_line**(*string*, *default*='black') – Color for the best threshold line.
- **lw_bestthresh_line**(*float*, *default*=2) – Linewidth for the best threshold line.
- **ls_bestthresh_line**(*string*, *default*='--') – Linestyle for the best threshold line.

References

CHAPTER 1

Indices and tables

- `genindex`
- `modindex`
- `search`

B

`BinaryClassification` (class in `plot_metric.functions`), 1

G

`get_function_parameters()`
(`plot_metric.functions.BinaryClassification`
`method`), 1

P

`plot_class_distribution()`
(`plot_metric.functions.BinaryClassification`
`method`), 1

`plot_confusion_matrix()`
(`plot_metric.functions.BinaryClassification`
`method`), 2

`plot_precision_recall_curve()`
(`plot_metric.functions.BinaryClassification`
`method`), 3

`plot_roc_curve()` (`plot_metric.functions.BinaryClassification`
`method`), 4

`plot_threshold()` (`plot_metric.functions.BinaryClassification`
`method`), 5