



UNIVERSIDADE
CATÓLICA
PORTUGUESA

BRAGA

Machine Learning

Session 8 - PL

Introduction to Supervised Learning

Degree in Applied Data Science

2024/2025

Classification Metrics with Scikit-Learn

- https://scikit-learn.org/stable/modules/model_evaluation.html

Scoring	Function	Comment
Classification		
'accuracy'	<code>metrics.accuracy_score</code>	
'balanced_accuracy'	<code>metrics.balanced_accuracy_score</code>	
'top_k_accuracy'	<code>metrics.top_k_accuracy_score</code>	
'average_precision'	<code>metrics.average_precision_score</code>	
'neg_brier_score'	<code>metrics.brier_score_loss</code>	
'f1'	<code>metrics.f1_score</code>	for binary targets
'f1_micro'	<code>metrics.f1_score</code>	micro-averaged
'f1_macro'	<code>metrics.f1_score</code>	macro-averaged
'f1_weighted'	<code>metrics.f1_score</code>	weighted average
'f1_samples'	<code>metrics.f1_score</code>	by multilabel sample
'neg_log_loss'	<code>metrics.log_loss</code>	requires <code>predict_proba</code> support
'precision' etc.	<code>metrics.precision_score</code>	suffixes apply as with 'f1'
'recall' etc.	<code>metrics.recall_score</code>	suffixes apply as with 'f1'
'jaccard' etc.	<code>metrics.jaccard_score</code>	suffixes apply as with 'f1'
'roc_auc'	<code>metrics.roc_auc_score</code>	
'roc_auc_ovr'	<code>metrics.roc_auc_score</code>	
'roc_auc_ovo'	<code>metrics.roc_auc_score</code>	
'roc_auc_ovr_weighted'	<code>metrics.roc_auc_score</code>	
'roc_auc_ovo_weighted'	<code>metrics.roc_auc_score</code>	

Regression Metrics with Scikit-Learn

- https://scikit-learn.org/stable/modules/model_evaluation.html

Regression	
'explained_variance'	<code>metrics.explained_variance_score</code>
'max_error'	<code>metrics.max_error</code>
'neg_mean_absolute_error'	<code>metrics.mean_absolute_error</code>
'neg_mean_squared_error'	<code>metrics.mean_squared_error</code>
'neg_root_mean_squared_error'	<code>metrics.root_mean_squared_error</code>
'neg_mean_squared_log_error'	<code>metrics.mean_squared_log_error</code>
'neg_root_mean_squared_log_error'	<code>metrics.root_mean_squared_log_error</code>
'neg_median_absolute_error'	<code>metrics.median_absolute_error</code>
'r2'	<code>metrics.r2_score</code>
'neg_mean_poisson_deviance'	<code>metrics.mean_poisson_deviance</code>
'neg_mean_gamma_deviance'	<code>metrics.mean_gamma_deviance</code>
'neg_mean_absolute_percentage_error'	<code>metrics.mean_absolute_percentage_error</code>
'd2_absolute_error_score'	<code>metrics.d2_absolute_error_score</code>
'd2_pinball_score'	<code>metrics.d2_pinball_score</code>
'd2_tweedie_score'	<code>metrics.d2_tweedie_score</code>

Error Estimation with Scikit-Learn

- Holdout:

- https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html

`sklearn.model_selection.train_test_split`

```
sklearn.model_selection.train_test_split(*arrays, test_size=None, train_size=None, random_state=None,  
shuffle=True, stratify=None) ¶
```

[\[source\]](#)

```
>>> X_train, X_test, y_train, y_test = train_test_split(  
...     X, y, test_size=0.33, random_state=42)  
...  
>>> X_train  
array([[4, 5],  
       [0, 1],  
       [6, 7]])  
>>> y_train  
[2, 0, 3]  
>>> X_test  
array([[2, 3],  
       [8, 9]])  
>>> y_test  
[1, 4]
```

Error Estimation with Scikit-Learn

- Cross Validation:

- https://scikit-learn.org/stable/modules/cross_validation.html#computing-cross-validated-metrics

`sklearn.model_selection.cross_val_score`

```
sklearn.model_selection.cross_val_score(estimator, X, y=None, *, groups=None, scoring=None, cv=None,
n_jobs=None, verbose=0, fit_params=None, params=None, pre_dispatch='2*n_jobs', error_score=nan)
```

[\[source\]](#)

```
>>> from sklearn.model_selection import cross_val_score
>>> clf = svm.SVC(kernel='linear', C=1, random_state=42)
>>> scores = cross_val_score(clf, X, y, cv=5)
>>> scores
array([0.96..., 1. , 0.96..., 0.96..., 1. ])
>>> print("%0.2f accuracy with a standard deviation of %0.2f" % (scores.mean(), scores.std()))
0.98 accuracy with a standard deviation of 0.02
```

Error Estimation with Scikit-Learn

- Others:
 - <https://scikit-learn.org/stable/modules/classes.html#splitter-classes>

<code>model_selection.GroupKFold([n_splits])</code>	K-fold iterator variant with non-overlapping groups.
<code>model_selection.GroupShuffleSplit(...)</code>	Shuffle-Group(s)-Out cross-validation iterator.
<code>model_selection.KFold([n_splits, shuffle, ...])</code>	K-Fold cross-validator.
<code>model_selection.LeaveOneGroupOut()</code>	Leave One Group Out cross-validator.
<code>model_selection.LeavePGroupsOut(n_groups)</code>	Leave P Group(s) Out cross-validator.
<code>model_selection.LeaveOneOut()</code>	Leave-One-Out cross-validator.
<code>model_selection.LeavePOut(p)</code>	Leave-P-Out cross-validator.
<code>model_selection.PredefinedSplit(test_fold)</code>	Predefined split cross-validator.
<code>model_selection.RepeatedKFold(*[, n_splits, ...])</code>	Repeated K-Fold cross validator.
<code>model_selection.RepeatedStratifiedKFold(*[, ...])</code>	Repeated Stratified K-Fold cross validator.
<code>model_selection.ShuffleSplit([n_splits, ...])</code>	Random permutation cross-validator.
<code>model_selection.StratifiedKFold([n_splits, ...])</code>	Stratified K-Fold cross-validator.
<code>model_selection.StratifiedShuffleSplit(...)</code>	Stratified ShuffleSplit cross-validator.
<code>model_selection.StratifiedGroupKFold(...)</code>	Stratified K-Fold iterator variant with non-overlapping groups.
<code>model_selection.TimeSeriesSplit([n_splits, ...])</code>	Time Series cross-validator.

Exercises:

- Notebooks on the github repository:
 - Notebook with examples:
 - `exercises/session08/examples.ipynb`
 - Notebook with exercises:
 - `exercises/session08/exercises.ipynb`