SCORING METRICS

SCORING METRIC

In machine learning, a scoring metric is the method by which a machine learning model is given a score.

There are many different ways to score machine learning models, hence many different scoring metrics.

REGRESSION: R2

The default classification scoring metric in `sklearn` is called “R2.”

R2, properly pronounced “R-squared,” is a measure of how well the data fits the regression line.

- 1.0 is perfect.
- 0.5 indicates half the data is explained by the line, half isn’t.
- 0 means none of the data is explained by the line.
- A negative score means the data runs opposite to the line!

REGRESSION: RMSE

RMSE, the Root Mean Squared Error, is a more common scoring metric.

The RMSE is the average distance that each point is from the regression line.

Example:

- Assume the Bike Rental dataset has an RMSE of 21.
- On average the predictions are 21 rentals off from the predictions.
- `sklearn` requires taking the negative square root of the `neg_mean_squared error` to get the RMSE.

ADDITIONAL REGRESSION SCORING METRICS

Regression	
'explained_variance'	`metrics.explained_variance_score`
'max_error'	`metrics.max_error`
'neg_mean_absolute_error'	`metrics.mean_absolute_error`
'neg_mean_squared_error'	`metrics.mean_squared_error`
'neg_root_mean_squared_error'	`metrics.mean_squared_error`
'neg_mean_squared_log_error'	`metrics.mean_squared_log_error`
'neg_median_absolute_error'	`metrics.median_absolute_error`
'r2'	`metrics.r2_score`
'neg_mean_poisson_deviance'	`metrics.mean_poisson_deviance`
'neg_mean_gamma_deviance'	`metrics.mean_gamma_deviance`
'neg_mean_absolute_percentage_error'	`metrics.mean_absolute_percentage_error`

CLASSIFICATION: ACCURACY

The default classification scoring metric is called “accuracy.”

Accuracy is the number of correct predictions divided by the number of total predictions:

- Accuracy = Correct / Total

IMBALANCED DATA

In imbalanced data, accuracy is not good enough.

Example:

- Assume that 1% of stars have detectable exoplanets.
- It’s easy to build a machine learning classifier with 99% accuracy.
- Just say that no stars have exoplanets!

POSITIVES & NEGATIVES

In machine learning, a positive case has a value of 1, and a negative case has a value of 0.

Example:

- Stars with exoplanets are positive (1% of data).
  - **True positives** - exoplanet stars correctly predicted.
  - **False positives** - exoplanet stars incorrectly predicted.

POSITIVES & NEGATIVES

Example Cont'd:

- Stars without exoplanets are negative (99% of data).
  - **True negatives** - stars without exoplanets correctly predicted.
  - **False negatives** - stars without exoplanets incorrectly predicted.

CLASSIFICATION: PRECISION

- The goal of precision is accuracy of positive predictions.
- Precision = True Pos / Total Pos
  - **Example:**
    - What percentage of predicted exoplanets were actually exoplanets?
    - Useful to publicize exoplanet verification.

CLASSIFICATION: RECALL

- The goal of recall is to find all the positive predictions.
- Recall = True Pos / (True Pos + False Neg)
  - **Example:**
    - Did we find all the exoplanet stars?
    - Useful to try and find all the exoplanets.

CLASSIFICATION: F1-SCORE

- The F1-Score is a balance between Precision and Recall.

ADDITIONAL CLASSIFICATION SCORING METRICS

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

HAPPY CODING!