twinlab.Emulator.score

Emulator.score(*params=<twinlab.params.ScoreParams object>, verbose=False*)

Score the performance of a trained emulator.

Returns a score for a trained emulator that quantifies its performance on the test dataset. Note that a test dataset must have been defined in order for this to produce a result. This means that `train_test_ratio` in TrainParams must be less than `1` when training the emulator. If there is no test dataset then this will return `None`. The score can be calculated using different metrics, see the `ScoreParams` class for a full list and description of available metrics.

Parameters:

- **params** (*ScoreParams, optional*) – A parameters object that contains optional scoring parameters.
- **verbose** (*bool, optional*) – Display detailed information about the operation while running.

Return type:

`Union`[`float`, `DataFrame`, `None`]

Returns:

Either a `pandas.DataFrame` containing the emulator per output dimension (if `combined_score = False`), or a `float` containing the combined score of the emulator averaged acorss output dimensions (if `combined_score = True`), or `None` if there was no test data defined during training.

Examples

Request the mean-standarised log loss (MSLL) averaged (combined) across all emulator output dimensions:

```
emulator = tl.Emulator("my_emulator")
params = tl.ScoreParams(metric="MSLL", combined_score=True)
emulator.score(params=params)
```

```
-4.07
```

Request the mean-squared error (MSE) for each output individually:

```
emulator = tl.Emulator("my_emulator")
params = tl.ScoreParams(metric="MSE", combined_score=False)
emulator.score(params=params)
```

```
pd.DataFrame({'y1': [1.8], 'y2': [0.9]})
```

Previous
twinlab.Emulator.summarise

Next
twinlab.Emulator.benchmark