[[evaluate]](https://keras.io/models/model/#evaluate)

to evaluate the model by getting test data accuracy

<model>.evaluate(x=None,

y=None,

batch\_size=None,

verbose=1,

sample\_weight=None,

steps=None,

callbacks=None)

* return:

Scalar test loss (if the model has a single output and no metrics) or list of scalars (if the model has multiple outputs and/or metrics). The attribute model.metrics\_names will give you the display labels for the scalar outputs.

* **x**: Numpy array of test data (if the model has a single input), or list of Numpy arrays (if the model has multiple inputs). If input layers in the model are named, you can also pass a dictionary mapping input names to Numpy arrays. x can be None (default) if feeding from framework-native tensors (e.g. TensorFlow data tensors).
* **y**: Numpy array of target (label) data (if the model has a single output), or list of Numpy arrays (if the model has multiple outputs). If output layers in the model are named, you can also pass a dictionary mapping output names to Numpy arrays. y can be None (default) if feeding from framework-native tensors (e.g. TensorFlow data tensors).
* **batch\_size**: Integer or None. Number of samples per evaluation step. If unspecified, batch\_size will default to 32.
* **verbose**: 0 or 1. Verbosity mode. 0 = silent, 1 = progress bar.
* **sample\_weight**: Optional Numpy array of weights for the test samples, used for weighting the loss function. You can either pass a flat (1D) Numpy array with the same length as the input samples (1:1 mapping between weights and samples), or in the case of temporal data, you can pass a 2D array with shape (samples, sequence\_length), to apply a different weight to every timestep of every sample. In this case you should make sure to specify sample\_weight\_mode="temporal" in compile().
* **steps**: Integer or None. Total number of steps (batches of samples) before declaring the evaluation round finished. Ignored with the default value of None.
* **callbacks**: List of keras.callbacks.Callback instances. List of callbacks to apply during evaluation. See callbacks.

[[predict]](https://keras.io/models/model/" \l "predict)

to use the model

<model>. predict(x,

batch\_size=None,

verbose=0,

steps=None,

callbacks=None)

* Return

Numpy array(s) of predictions.

* **x**: The input data, as a Numpy array (or list of Numpy arrays if the model has multiple inputs).
* **batch\_size**: Integer. If unspecified, it will default to 32.
* **verbose**: Verbosity mode, 0 or 1.
* **steps**: Total number of steps (batches of samples) before declaring the prediction round finished. Ignored with the default value of None.
* **callbacks**: List of keras.callbacks.Callback instances. List of callbacks to apply during prediction. See callbacks.