

## graphlab.nearest\_neighbor\_classifier.NearestNeighborClassifier.classify

`NearestNeighborClassifier.classify(dataset, max_neighbors=10, radius=None, verbose=True)`

Return the predicted class for each observation in `dataset`, based on the closest neighbors stored in the nearest neighbors classifier model.

**Parameters:** **dataset** : SFrame

Dataset of new observations. Must include columns with the same names as the features used for model training, but does not require a target column. Additional columns are ignored.

**verbose** : bool, optional

If True, print progress updates.

**max\_neighbors** : int, optional

Maximum number of neighbors to consider for each point.

**radius** : float, optional

Maximum distance from each point to a neighbor in the reference dataset.

**Returns:** **out** : SFrame

An SFrame with model predictions. The first column is the most likely class according to the model, and the second column is the predicted probability for that class.

#### See also

`create`, `predict`, `predict_topk`

#### Notes

- If the 'radius' parameter is small, it is possible that a query point has no qualified neighbors in the training dataset. In this case, the resulting class and probability for that query are 'None' in the SFrame output by this method. If the target column in the training dataset has missing values, these predictions will be ambiguous.
- Ties between predicted classes are broken randomly.

#### Examples

```
>>> sf_train = graphlab.SFrame({'species': ['cat', 'dog', 'fossa', 'dog'],
...                               'height': [9, 25, 20, 23],
...                               'weight': [13, 28, 33, 22]})
>>> sf_new = graphlab.SFrame({'height': [26, 19],
...                               'weight': [25, 35]})
>>> m = graphlab.nearest_neighbor_classifier.create(sf, target='species')
>>> ystar = m.classify(sf_new, max_neighbors=2)
>>> print ystar
+-----+-----+
| class | probability |
+-----+-----+
| dog   | 1.0         |
| fossa | 0.5         |
+-----+-----+
```

