Semi-supervised learning

Have you heard of supervised learning and unsupervised learning?

Supervised learning in simple terms is learning by examples. The instructor shows you the object (data) and tells you the true label associated with the object. You learn the mappings (distribution) of the data for the labels and the instructor corrects you (loss minimisation) when you make a mistake.

On the contrary, if the instructor only provides you with the object (data) without any associated label, you group the objects based on similar characteristics using a distance metric in the feature space. The groups thus formed are highly uncertain. This is **unsupervised learning**

However, what if the instructor gives the true labels for a very small set of those objects?

Using supervised learning, would not be much useful as the feedback is too sparse, due to low number of true labels. The distribution mapping for objects associated with the true labels is insufficient to generalise.

Semi-supervised learning comes to the rescue, wherein we learn the generalised distribution from the entire collection of objects and then associate the objects to the closest neighbour based on the feedback received.

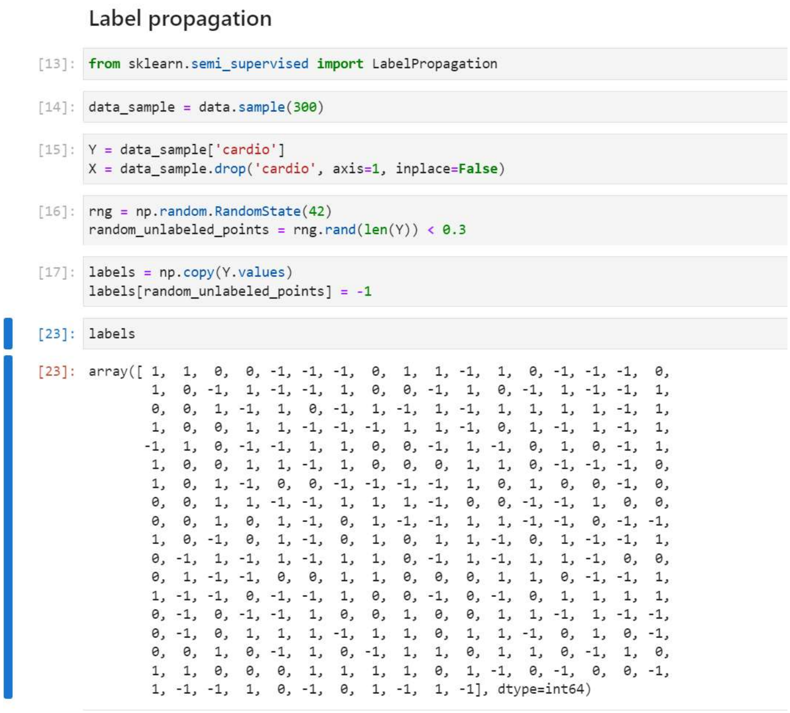
"The intuition for the broader approach of semi-supervised learning is that nearby points in the input space should have the same label, and points in the same structure or manifold in the input space should have the same label" --- Jason Brownlee

How do we apply semi-supervised learning using scikit-learn?

Refer to this blog https://machinelearningmastery.com/semi-supervised-learning-with-label-spreading/

Just as we use scikit-learn apis to train any classification model using "fit()", semi-supervised learning models are trained with the basic difference in the data preparation as:

We have the data split as X_train with corresponding y_train labels, X_unlabeled which does not have the associated labels and a test set. The y_train labels contains encoded class labels with unlabeled points having -1 for their labels.



Label propagation

```
[13]: from sklearn.semi_supervised import LabelPropagation

[14]: data_sample = data.sample(300)

[15]: Y = data_sample['cardio']
      X = data_sample.drop('cardio', axis=1, inplace=False)

[16]: rng = np.random.RandomState(42)
      random_unlabeled_points = rng.rand(len(Y)) < 0.3

[17]: labels = np.copy(Y.values)
      labels[random_unlabeled_points] = -1

[23]: labels

[23]: array([ 1,  1,  0,  0, -1, -1, -1,  0,  1,  1, -1,  1,  0, -1, -1, -1,  0,
              1,  0, -1,  1, -1, -1,  1,  0,  0, -1,  1,  0, -1,  1, -1, -1,  1,
              0,  0,  1, -1,  1,  0, -1,  1, -1,  1, -1,  1,  1,  1,  1, -1,  1,
              1,  0,  0,  1,  1, -1, -1, -1,  1,  1, -1,  0,  1, -1,  1, -1,  1,
             -1,  1,  0, -1, -1,  1,  1,  0,  0, -1,  1, -1,  0,  1,  0, -1,  1,
              1,  0,  0,  1,  1, -1,  1,  0,  0,  0,  1,  1,  0, -1, -1, -1,  0,
              1,  0,  1, -1,  0,  0, -1, -1, -1, -1,  1,  0,  1,  0,  0, -1,  0,
              0,  0,  1,  1, -1, -1,  1,  1,  1, -1,  0,  0, -1, -1,  1,  0,  0,
              0,  0,  1,  0,  1, -1,  0,  1, -1, -1,  1,  1, -1, -1,  0, -1, -1,
              1,  0, -1,  0,  1, -1,  0,  1,  0,  1,  1, -1,  0,  1, -1, -1,  1,
              0, -1,  1, -1,  1, -1,  1,  1,  0, -1,  1, -1,  1,  1, -1,  0,  0,
              0,  1, -1, -1,  0,  0,  1,  1,  0,  0,  0,  1,  1,  0, -1, -1,  1,
              1, -1, -1,  0, -1, -1,  1,  0,  0, -1,  0, -1,  0,  1,  1,  1,  1,
              0, -1,  0, -1, -1,  1,  0,  0,  1,  0,  0,  1,  1, -1,  1, -1, -1,
              0, -1,  0,  1,  1,  1, -1,  1,  1,  0,  1,  1, -1,  0,  1,  0, -1,
              0,  0,  1,  0, -1,  1,  0, -1,  1,  1,  0,  1,  1,  0, -1,  1,  0,
              1,  1,  0,  0,  0,  1,  1,  1,  1,  0,  1, -1,  0, -1,  0,  0, -1,
              1, -1, -1,  1,  0, -1,  0,  1, -1,  1, -1], dtype=int64)
```

Here we prepared a dataset with 30% unlabeled data points. The unlabeled data points are associated with -1 as the class label

Apart from label propagation, scikit-learn library provides another interesting semi-supervised algorithm “label spreading”, having the same underlying principle as label propagation with the difference in how the similarity matrix is updated.

Refer: https://scikit-learn.org/stable/modules/semi_supervised.html#label-propagation

What are the different approaches to Semi-supervised learning?

1. Self-training
2. Graph based
3. Low density separation

Label spreading and label propagation are modifications of graph based Semi-supervised learning

Let us briefly understand the 3 approaches

1. In self-training we first train a classification model on labelled data alone. Use this model to label a sample of data points from unlabeled set. Use the combined set of labelled and pseudo labelled data points and retrain the model. Repeat the process iteratively

2. All data is represented by a vertex in a graph. The edges with associated weights signify the closeness of two vertices. The method learns the distribution mapping of the data and based on the idea of proximal points belonging to the same category, the algorithm is designed. [Label spreading and label propagation are graph based semi-supervised learning]

3. It is based on a very simple idea that the decision boundary should lie in the low density region in the feature space

References:

https://www.javatpoint.com/semi-supervised-learning [Underlying assumptions of Semi-supervised learning]

https://pypi.org/project/semisupervised/ [SVM based semi-supervised learning models]

https://pythonawesome.com/a-pytorch-based-library-for-semi-supervised-learning/ [Pytorch based semi supervised learning models]

https://www.finsliqblog.com/ai-and-machine-learning/types-of-semi-supervised-algorithms/ [Types of semi-supervised learning and case studies]