**Self-Training Classifier: How to Make Any Algorithm Behave**

**Like a Semi-Supervised One**

Semi-Supervised Learning combines labeled and unlabeled examples to expand the available data pool for model training. As a result, we can improve model performance and save a lot of time and money by not having to label thousands of examples manually.

If you have your favorite **Supervised** Machine Learning algorithm, you will be happy to hear that you can quickly adapt it to use a **Semi-Supervised** approach through a technique called **Self-Training**.

# Intro

Semi-Supervised Learning combines labeled and unlabeled examples to expand the available data pool for model training. As a result, we can improve model performance and save a lot of time and money by not having to label thousands of examples manually.

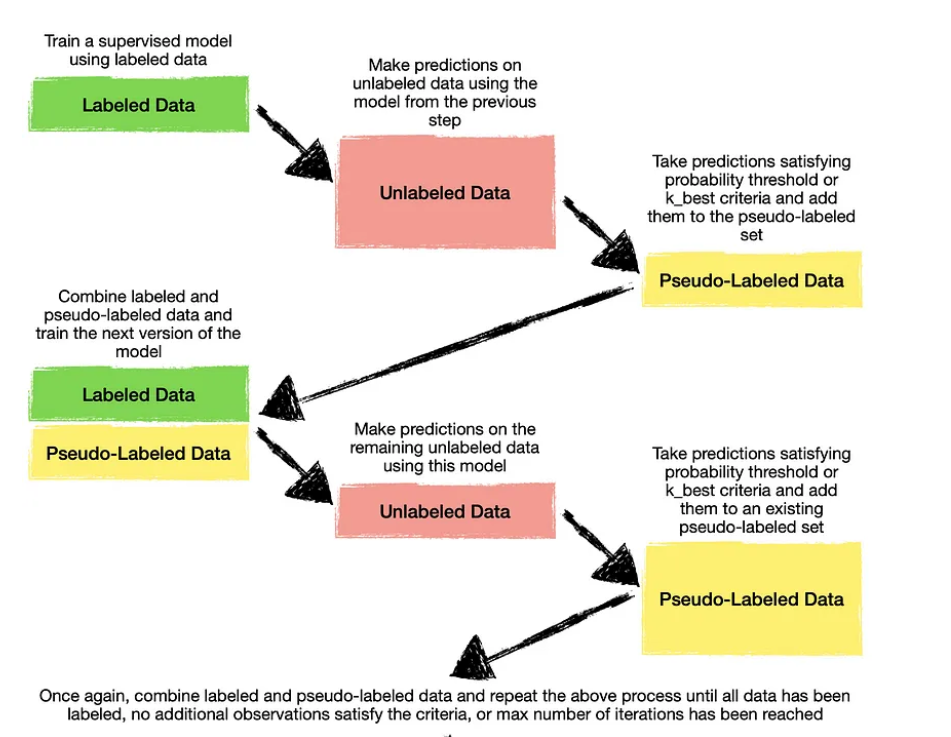
If you have your favorite **Supervised** Machine Learning algorithm, you will be happy to hear that you can quickly adapt it to use a **Semi-Supervised** approach through a technique called **Self-Training**.

# How does Self-Training work?

You may think that Self-Training involves some magic or uses a highly complex approach. In reality, though, the idea behind Self-Training is very straightforward and can be explained by the following steps:

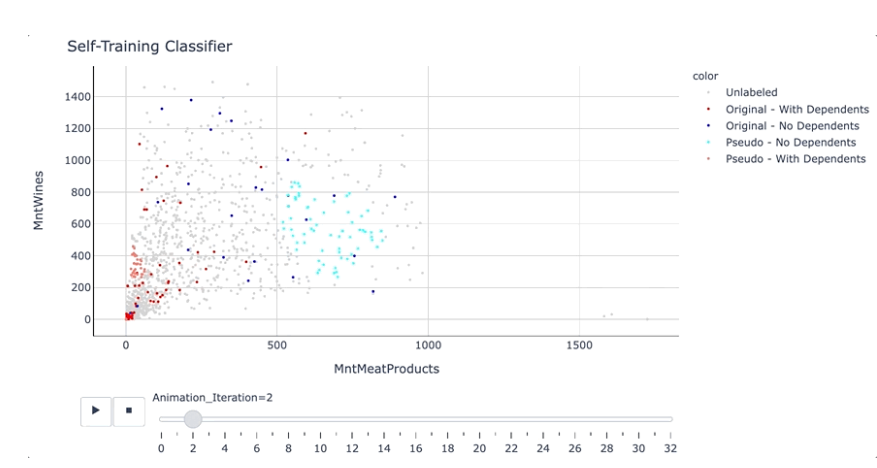
1. First, we gather all labeled and unlabeled data, but we **only use labeled observations to train** our first supervised model.
2. Then we use this model to predict the class of unlabeled data.
3. In the third step, we select observations that satisfy our predefined criteria (e.g., prediction probability is >90% or belongs to the top 10 of observations with the highest prediction probabilities) and **combine these pseudo-labels** with labeled data.
4. We repeat the process by training **a new supervised model** using observations **with labels and pseudo-labels**. Then we make predictions again and add newly selected observations into the pseudo-labeled pool.
5. We **iterate through these steps until** we finish labeling all the data, no additional unlabeled observations satisfy our pseudo-labeling criteria, or we reach the specified max number of iterations.

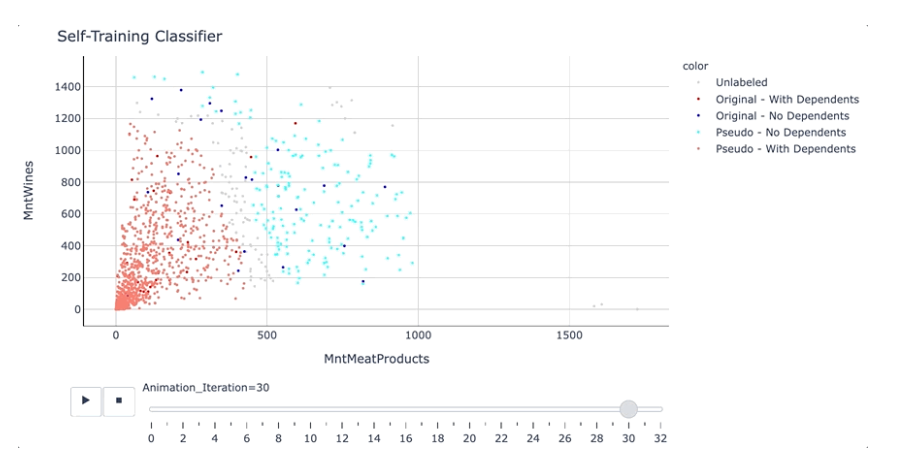
Here is an illustration that summarizes all of the steps I have just described:

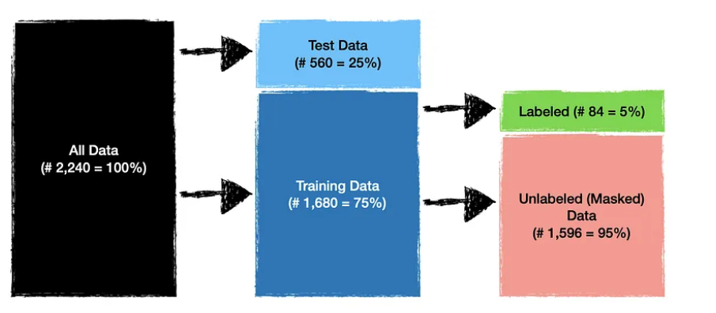




k-best criteria belong to the top 10 of observations with the highest prediction probabilities whether they do not satisfy the threshold.







# The difference between Self-Training Classifier is that you need to use a classifier model to label the unlabeled data. Noted that for label propagation and label spreading we do not need to use another model to label the unlabeled data.

**Parameters:**

<https://scikit-learn.org/stable/modules/generated/sklearn.semi_supervised.SelfTrainingClassifier.html>

**base\_estimator *estimator object***

An estimator object implementing fit and predict\_proba. Invoking the fit method will fit a clone of the passed estimator, which will be stored in the base\_estimator\_ attribute.

**Threshold *float, default=0.75***

The decision threshold for use with criterion='threshold'. Should be in [0, 1). When using the 'threshold' criterion, a [well calibrated classifier](https://scikit-learn.org/stable/modules/calibration.html#calibration) should be used.

**Criterion *{‘threshold’, ‘k\_best’}, default=’threshold’***

The selection criterion used to select which labels to add to the training set. If 'threshold', pseudo-labels with prediction probabilities above threshold are added to the dataset. If 'k\_best', the k\_best pseudo-labels with highest prediction probabilities are added to the dataset. When using the ‘threshold’ criterion, a [well calibrated classifier](https://scikit-learn.org/stable/modules/calibration.html#calibration) should be used.

**k\_best *int, default=10***

The amount of samples to add in each iteration. Only used when criterion='k\_best'.

**max\_iter *int or None, default=10***

Maximum number of iterations allowed. Should be greater than or equal to 0. If it is None, the classifier will continue to predict labels until no new pseudo-labels are added, or all unlabeled samples have been labeled.

**Verbose *bool, default=False***

Enable verbose output.