

Appendix A. Adding a multiview classifier

In order to plug a multi-view classifier, the user has to add the following python file named after the classifier in the `multiview_classifiers` folder, and the classifier will be automatically added to the available pool.

```

1 from new_mv_algo_module import NewMVAlgo
2 from ..multiview.multiview_utils import BaseMultiviewClassifier
3
4 from ..utils.hyper_parameter_search import CustomRandint
5
6 classifier_class_name = "NewMVAlgoClassifier"
7
8 class NewMVAlgoClassifier(BaseMultiviewClassifier, NewMVAlgo):
9
10     def __init__(self, param_1=50,
11                  random_state=None,
12                  param_2="edge"):
13         BaseMultiviewClassifier.__init__(self, random_state)
14         NewMVAlgo.__init__(self, param_1=param_1,
15                             random_state=random_state,
16                             param_2=param_2)
17         self.param_names = ["param_1", "random_state", "param_2"]
18         self.distribs = [CustomRandint(5,200), [random_state], ["val_1",
19 "val_2"]]
20
21     def fit(self, X, y, train_indices=None, view_indices=None):
22         # This function is used to initialize the sample and view indices, in
23         # case they are None, it transforms them in the correct values
24         train_indices, view_indices = get_samples_views_indices(X,
25                                                                 train_indices,
26                                                                 view_indices)
27         needed_input = transform_data_if_needed(X, train_indices, view_indices)
28         return NewMVAlgo.fit(self, needed_input, y[train_indices])
29
30     def predict(self, X, sample_indices=None, view_indices=None):
31         sample_indices, view_indices = get_samples_views_indices(X,
32                                                                 sample_indices,
33                                                                 view_indices)
34         needed_input = transform_data_if_needed(X, sample_indices, view_indices)
35         return NewMVAlgo.predict(self, needed_input)

```

Appendix B. Adding a monoview classifier

Similarly, to plug a mono-view classifier, the user has to add the following python file named after the classifier in the `monoview_classifiers` folder, and the classifier will be automatically added to the available pool.

```

1 import Algo
2 from ..monoview.monoview_utils import BaseMonoviewClassifier, CustomUniform,
   CustomRandint
3
4 classifier_class_name = "AlgoClassifier"
5
6 class AlgoClassifier(Algo, BaseMonoviewClassifier):
7
8     def __init__(self, random_sate=42, trade_off=0.5, norm_type='l1',
   max_depth=50)
9
10        super(AlgoClassifier, self).__init__(random_sate=random_sate,
11                                             trade_off=trade_off,
12                                             norm_type=norm_type,
13                                             max_depth=max_depth)
14
15        self.param_names = ["trade_off", "norm_type", "max_depth"]
16        self.distribs = [CustomUniform(),
17                        ["l1", "l2"],
18                        CustomRandint()]

```