
mlstacking

Release 0.2.0

May 27, 2019

Contents:

1	Installation Guide	3
1.1	Installation	3
2	Python API	5
2.1	Class	5
2.2	Example	5
	Python Module Index	7

mlstacking is a Python module for sklearn-API friendly multi-layer stacking built on top of scikit-learn and distributed under the 3-Clause BSD license.

1.1 Installation

1.1.1 Dependencies

mlstacking requires:

- Python (≥ 3.5)
- NumPy ($\geq 1.11.0$)
- SciPy ($\geq 0.17.0$)
- joblib (≥ 0.11)

1.1.2 User installation

If you already have a working installation of numpy and scipy, the easiest way to install mlstacking is using pip.

```
pip install mlstacking
```


2.1 Class

Scikit-Learn Wrapper interface for multi-layer stacking.

```
class mlstacking.sklearn.StackingModel (base_models, meta_model, pre-  
                                         dict_mode='average', n_folds=5,  
                                         keep_layer_results=True)
```

Implementation of the Scikit-Learn API for multi-layer stacking.

Parameters

- **base_models** (*list*) – List of list of sklearn type classifiers
- **meta_model** (*object*) – Sklearn type classifiers
- **predict_mode** (*string*) – Specify which predict to use: average, once
- **n_folds** (*int*) – Depend how many folds each classifier run
- **keep_layer_results** (*boolean*) – Keep results of each layer or not

2.2 Example

```
import numpy
from mlstacking.sklearn import StackingModel
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from xgboost.sklearn import XGBClassifier

X = numpy.random.rand(10,10)
Y = numpy.random.randint(0,2,(10,1))

base_models = [[DecisionTreeClassifier(),RandomForestClassifier()],
```

(continues on next page)

(continued from previous page)

```
[RandomForestClassifier(),XGBClassifier(),],
[XGBClassifier(),DecisionTreeClassifier(),],]

sm = StackingModel(base_models,XGBClassifier())
sm.fit(X,Y)

sm.predict(X)
# array([0, 0, 0, 1, 0, 0, 0, 1, 1, 1])

sm.predict_proba(X)
# array([[0.6039953 , 0.39600468],
#        [0.6039953 , 0.39600468],
#        [0.6039953 , 0.39600468],
#        [0.40033996, 0.59966004],
#        [0.6039953 , 0.39600468],
#        [0.6039953 , 0.39600468],
#        [0.6039953 , 0.39600468],
#        [0.40033996, 0.59966004],
#        [0.40033996, 0.59966004],
#        [0.40033996, 0.59966004]], dtype=float32)
```

m

`mlstacking.sklearn`, [5](#)

M

`mlstacking.sklearn` (*module*), [5](#)

S

`StackingModel` (*class in mlstacking.sklearn*), [5](#)