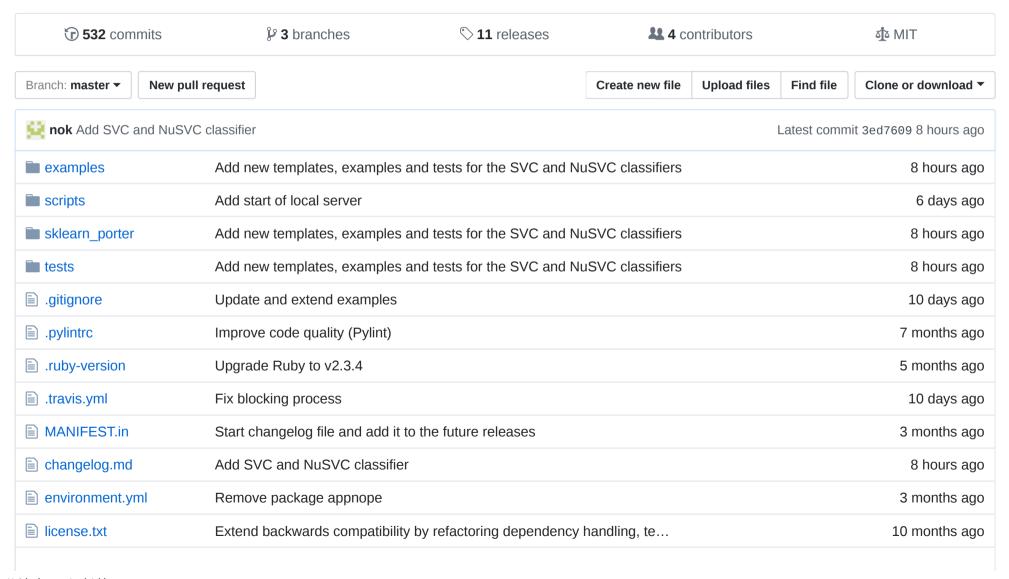
#### nok / sklearn-porter

Transpile trained scikit-learn estimators to C, Java, JavaScript and others.

#scikit-learn #machine-learning #data-science #sklearn

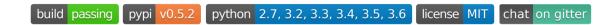


https://github.com/nok/sklearn-porter 1/9

readme.md	Add new templates, examples and tests for the SVC and NuSVC classifiers	8 hours ago
requirements.txt	Extend backwards compatibility by refactoring dependency handling, te	10 months ago
setup.cfg	Add license file	10 months ago
setup.py	Refactor and fix lost requirements.txt file	3 months ago

**readme.md** 

# sklearn-porter



Transpile trained scikit-learn estimators to C, Java, JavaScript and others.

It's recommended for limited embedded systems and critical applications where performance matters most.

## **Machine learning algorithms**

Algorithm	Programming language							
Classification	Java *	JS	С	Go	PHP	Ruby		
svm.SVC	√, √ 1	<b>√</b>	<b>√</b>		<b>√</b>	<b>√</b>		
svm.NuSVC	√, √ 1	<b>√</b>	<b>√</b>		<b>√</b>	<b>√</b>		
svm.LinearSVC	<b>√</b>	<b>√</b>	<b>√</b>	<b>√</b>	<b>√</b>	<b>√</b>		
tree.DecisionTreeClassifier	√, √ <sup>E</sup> , √ <sup>I</sup>	√, √ E						
ensemble.RandomForestClassifier	√ E, √ I	√ E	√ E		√ E	√ E		

https://github.com/nok/sklearn-porter 2/9

ensemble.ExtraTreesClassifier	√ E, √ I	√ E	√ E	√ E	√ E
ensemble.AdaBoostClassifier	√ E, √ I	√ E, √ I	√ E		
neighbors.KNeighborsClassifier	√, √ ¹	√, √ ¹			
naive_bayes.GaussianNB	<b>√</b>	<b>√</b>			
naive_bayes.BernoulliNB	<b>√</b>	<b>√</b>			
neural_network.MLPClassifier	√, √ ¹	<b>√</b>			
Regression					
neural_network.MLPRegressor		✓			

 $\sqrt{\ }$  = is full-featured,  $^{\rm E}$  = with embedded model data,  $^{\rm I}$  = with imported model data,  $^{\rm A}$  = default language

## Installation

```
$ pip install sklearn-porter
```

If you want the latest changes, you can install the module from the master branch:

```
$ pip uninstall -y sklearn-porter
$ pip install --no-cache-dir https://github.com/nok/sklearn-porter/zipball/master
```

## **Minimum requirements**

https://github.com/nok/sklearn-porter 3/9

```
- python>=2.7.3
- scikit-learn>=0.14.1
```

If you want to transpile a multilayer perceptron, you have to upgrade the scikit-learn package:

```
- python>=2.7.3
- scikit-learn>=0.18.0
```

### **Usage**

#### **Export**

The following example shows how you can port a decision tree estimator to Java:

```
from sklearn.datasets import load_iris
from sklearn.tree import tree
from sklearn_porter import Porter

# Load data and train the classifier:
samples = load_iris()
X, y = samples.data, samples.target
clf = tree.DecisionTreeClassifier()
clf.fit(X, y)

# Export:
porter = Porter(clf, language='java')
output = porter.export(embed_data=True)
print(output)
```

The exported result matches the official human-readable version of the decision tree.

https://github.com/nok/sklearn-porter 4/9

#### **Prediction**

Run the prediction(s) in the target programming language directly:

```
# ...
porter = Porter(clf, language='java')

# Prediction(s):
Y_java = porter.predict(X)
y_java = porter.predict(X[0])
y_java = porter.predict([1., 2., 3., 4.])
```

### **Integrity**

Always compute and test the integrity between the original and the transpiled estimator:

```
# ...
porter = Porter(clf, language='java')

# Accuracy:
integrity = porter.integrity_score(X)
print(integrity) # 1.0
```

#### **Command-line interface**

First of all have a quick view on the available arguments:

https://github.com/nok/sklearn-porter 5/9

The following example shows how you can save an trained estimator to the pickle format:

```
# ...
# Extract estimator:
joblib.dump(clf, 'estimator.pkl')
```

After that the estimator can be transpiled to JavaScript by using the following command:

```
$ python -m sklearn_porter -i estimator.pkl --js
```

The target programming language is changeable on the fly:

```
$ python -m sklearn_porter -i estimator.pkl --c
$ python -m sklearn_porter -i estimator.pkl --go
$ python -m sklearn_porter -i estimator.pkl --php
$ python -m sklearn_porter -i estimator.pkl --java
$ python -m sklearn_porter -i estimator.pkl --ruby
```

For further processing the argument --pipe can be used to pass the result:

```
$ python -m sklearn_porter -i estimator.pkl --js --pipe > estimator.js
```

For instance the result can be minified by using UglifyJS:

```
$ python -m sklearn_porter -i estimator.pkl --js --pipe | uglifyjs --compress -o estimator.min.js
```

Further information will be shown by using the --help argument:

https://github.com/nok/sklearn-porter

6/9

```
$ python -m sklearn_porter --help
$ python -m sklearn_porter -h
```

## **Development**

#### **Environment**

Install the required environment modules by executing the script environment.sh:

```
$ bash ./scripts/environment.sh

#!/usr/bin/env bash

conda env create -c conda-forge -n sklearn-porter python=2 -f environment.yml
source activate sklearn-porter
```

The following compilers or intepreters are required to cover all tests:

- GCC (>=4.2)
- Java ( >=1.6 )
- PHP ( >=7 )
- Ruby ( >=2.4.1 )
- Go (>=1.7.4)
- Node.js ( >=6 )

#### **Testing**

The tests cover module functions as well as matching predictions of transpiled estimators. Run all tests by executing the script test.sh:

```
$ bash ./scripts/test.sh

#!/usr/bin/env bash

python -m unittest discover -vp '*Test.py'

The test files have a specific pattern: '[Algorithm][Language]Test.py':

$ python -m unittest discover -vp 'RandomForest*Test.py'
$ python -m unittest discover -vp '*JavaTest.py'
```

While you are developing new features or fixes, you can reduce the test duration by setting the number of tests:

```
$ N_RANDOM_FEATURE_SETS=15 N_EXISTING_FEATURE_SETS=30 python -m unittest discover -vp '*Test.py'
```

#### Quality

It's highly recommended to ensure the code quality. For that I use Pylint, which you can run by executing the script lint.sh:

```
$ bash ./scripts/lint.sh

#!/usr/bin/env bash
find ./sklearn_porter -name '*.py' -exec pylint {} \;
```

https://github.com/nok/sklearn-porter

### Citation

If you use this implementation in you work, please add a reference/citation to the paper. You can use the following BibTeX entry:

```
@misc{SkPoDaMo,
   author = {Darius Morawiec},
   title = {sklearn-porter: Transpile trained scikit-learn estimators to C, Java, JavaScript and others},
   url = {https://github.com/nok/sklearn-porter},
   year = {2016--2017}
}
```

### License

The module is Open Source Software released under the MIT license.

## **Questions?**

Don't be shy and feel free to contact me on Twitter or Gitter.

https://github.com/nok/sklearn-porter 9/9