

3.4. Model persistence

After training a scikit-learn model, it is desirable to have a way to persist the model for future use without having to retrain. The following section gives you an example of how to persist a model with pickle. We'll also review a few security and maintainability issues when working with pickle serialization.

An alternative to pickling is to export the model to another format using one of the model export tools listed under [Related Projects](#). Unlike pickling, once exported you cannot recover the full Scikit-learn estimator object, but you can deploy the model for prediction, usually by using tools supporting open model interchange formats such as [ONNX](#) or [PMML](#).

3.4.1. Persistence example

It is possible to save a model in scikit-learn by using Python's built-in persistence model, namely [pickle](#):

```
>>> from sklearn import svm
>>> from sklearn import datasets
>>> clf = svm.SVC()
>>> X, y = datasets.load_iris(return_X_y=True)
>>> clf.fit(X, y)
SVC()

>>> import pickle
>>> s = pickle.dumps(clf)
>>> clf2 = pickle.loads(s)
>>> clf2.predict(X[0:1])
array([0])
>>> y[0]
0
```

In the specific case of scikit-learn, it may be better to use joblib's replacement of pickle (`dump` & `load`), which is more efficient on objects that carry large numpy arrays internally as is often the case for fitted scikit-learn estimators, but can only pickle to the disk and not to a string:

```
>>> from joblib import dump, load
>>> dump(clf, 'filename.joblib')
```

Later you can load back the pickled model (possibly in another Python process) with:

```
>>> clf = load('filename.joblib')
```

Note: `dump` and `load` functions also accept file-like object instead of filenames. More information on data persistence with Joblib is available [here](#).

3.4.2. Security & maintainability limitations

pickle (and joblib by extension), has some issues regarding maintainability and security. Because of this,

- Never unpickle untrusted data as it could lead to malicious code being executed upon loading.
- While models saved using one version of scikit-learn might load in other versions, this is entirely unsupported and inadvisable. It should also be kept in mind that operations performed on such data could give different and unexpected results.

In order to rebuild a similar model with future versions of scikit-learn, additional metadata should be saved along the pickled model:

- The training data, e.g. a reference to an immutable snapshot
- The python source code used to generate the model
- The versions of scikit-learn and its dependencies
- The cross validation score obtained on the training data

This should make it possible to check that the cross-validation score is in the same range as before.

Since a model internal representation may be different on two different architectures, dumping a model on one architecture and loading it on another architecture is not supported.

If you want to know more about these issues and explore other possible serialization methods, please refer to this [talk by Alex Gaynor](#).

Toggle Menu

© 2007 - 2019, scikit-learn developers (BSD License). [Show this page source](#)