

```
(base) PS C:\Users\Abi Rahman> python
```

```
Python 3.10.12 | packaged by Anaconda, Inc. | (main, Jul 5 2023, 19:01:18) [MSC v.1916 64 bit  
(AMD64)] on win32
```

```
Type "help", "copyright", "credits" or "license" for more information.
```

```
>>> from sklearn import datasets
```

```
>>> iris = datasets.load_iris()
```

```
>>> digits = datasets.load_digits()
```

```
>>> print(digits.data)
```

```
[[ 0.  0.  5. ...  0.  0.  0.]
```

```
 [ 0.  0.  0. ... 10.  0.  0.]
```

```
 [ 0.  0.  0. ... 16.  9.  0.]
```

```
...
```

```
 [ 0.  0.  1. ...  6.  0.  0.]
```

```
 [ 0.  0.  2. ... 12.  0.  0.]
```

```
 [ 0.  0. 10. ... 12.  1.  0.]]
```

```
>>>
```

```
>>> digits.target
```

```
array([0, 1, 2, ..., 8, 9, 8])
```

```
>>>
```

```
>>> #Shape of the data arrays
```

```
>>> digits.images[0]
```

```
array([[ 0.,  0.,  5., 13.,  9.,  1.,  0.,  0.],
```

```
       [ 0.,  0., 13., 15., 10., 15.,  5.,  0.],
```

```
       [ 0.,  3., 15.,  2.,  0., 11.,  8.,  0.],
```

```
       [ 0.,  4., 12.,  0.,  0.,  8.,  8.,  0.],
```

```
       [ 0.,  5.,  8.,  0.,  0.,  9.,  8.,  0.],
```

```
       [ 0.,  4., 11.,  0.,  1., 12.,  7.,  0.],
```

```
       [ 0.,  2., 14.,  5., 10., 12.,  0.,  0.],
```

```
       [ 0.,  0.,  6., 13., 10.,  0.,  0.,  0.]])
```

```
>>>
```

```
>>> #Loading from external datasets
```

```

>>> from sklearn import svm
>>> clf = svm.SVC(gamma=0.001, C=100.)
>>>
>>> #Choosing the parameters of the model
>>> clf.fit(digits.data[:-1], digits.target[:-1])
SVC(C=100.0, gamma=0.001)
>>> clf.predict(digits.data[-1:])
array([8])
>>>
>>>
>>> #Conventions
>>> #Type casting
>>> import numpy as np
>>> from sklearn import kernel_approximation
>>> rng = np.random.RandomState(0)
>>> X = rng.rand(10, 2000)
>>> X = np.array(X, dtype='float32')
>>> X.dtype
dtype('float32')
>>>
>>> transformer = kernel_approximation.RBFSampler()
>>> X_new = transformer.fit_transform(X)
>>> X_new.dtype
dtype('float32')
>>>
>>>
>>> #Regression targets are cast to float64 and classification targets are maintained:
>>> from sklearn import datasets
>>> from sklearn.svm import SVC
>>> iris = datasets.load_iris()
File "<stdin>", line 1

```

```

iris = datasets.load_iris()
IndentationError: unexpected indent
>>> iris = datasets.load_iris()
>>> clf = SVC()
>>> clf.fit(iris.data, iris.target)
SVC()
>>> list(clf.predict(iris.data[:3]))
[0, 0, 0]
>>> clf.fit(iris.data, iris.target_names[iris.target])
SVC()
>>> list(clf.predict(iris.data[:3]))
['setosa', 'setosa', 'setosa']
>>>
>>>
>>> #Refitting and updating parameters
>>> import numpy as np
>>> from sklearn.datasets import load_iris
>>> from sklearn.svm import SVC
>>> X, y = load_iris(return_X_y=True)
>>> clf = SVC()
>>> clf.set_params(kernel='linear').fit(X, y)
SVC(kernel='linear')
>>> clf.predict(X[:5])
array([0, 0, 0, 0, 0])
>>> clf.set_params(kernel='rbf').fit(X, y)
SVC()
>>> clf.predict(X[:5])
array([0, 0, 0, 0, 0])
>>>
>>>
>>> #Multiclass vs. multilabel fitting

```

```
>>> from sklearn.multiclass import OneVsRestClassifier
>>> from sklearn.preprocessing import LabelBinarizer
>>> X = [[1, 2], [2, 4], [4, 5], [3, 2], [3, 1]]
>>> y = [0, 0, 1, 1, 2]
>>> classif = OneVsRestClassifier(estimator=SVC(random_state=0))
>>> classif.fit(X, y).predict(X)
array([0, 0, 1, 1, 2])
>>>
>>> y = LabelBinarizer().fit_transform(y)
>>> classif.fit(X, y).predict(X)
array([[1, 0, 0],
       [1, 0, 0],
       [0, 1, 0],
       [0, 0, 0],
       [0, 0, 0]])
>>>
>>> from sklearn.preprocessing import MultiLabelBinarizer
>>> y = [[0, 1], [0, 2], [1, 3], [0, 2, 3], [2, 4]]
>>> y = MultiLabelBinarizer().fit_transform(y)
>>> classif.fit(X, y).predict(X)
array([[1, 1, 0, 0, 0],
       [1, 0, 1, 0, 0],
       [0, 1, 0, 1, 0],
       [1, 0, 1, 0, 0],
       [1, 0, 1, 0, 0]])
>>>
```