

```
In [29]: from sklearn.ensemble import RandomForestClassifier
clf = RandomForestClassifier(random_state=0)
X=[[1, 2, 3],[11, 12, 13]]
y=[0,1]
clf.fit(X,y)
RandomForestClassifier(random_state=0)
clf.predict(X)
clf.predict([[4, 5, 6],[14, 15, 16]])
```

Out[29]: array([0, 1])

```
In [34]: from sklearn.preprocessing import StandardScaler
X=[[0, 15],[1, -10]]
StandardScaler().fit(X).transform(X)
```

Out[34]: array([[ -1., 1.],
 [ 1., -1.]])

```
In [32]: from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.pipeline import make_pipeline
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

pipe = make_pipeline(
    StandardScaler(),
    LogisticRegression(random_state=0)
)
X,y = load_iris(return_X_y=True)
X_train,X_test,y_train,y_test = train_test_split(X,y,random_state=0)
pipe.fit(X_train,y_train)
accuracy_score(pipe.predict(X_test),y_test)
```

Out[32]: 0.9736842105263158

```
In [27]: from sklearn.datasets import make_regression
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import cross_validate

X,y = make_regression(n_samples=1000, random_state=0)
lr=LinearRegression()

result = cross_validate(lr,X,y)
result['test_score']
```

Out[27]: array([1., 1., 1., 1., 1.])

```
In [47]: 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)

clf.predict(X[:5])

clf.set_params(kernel='rbf').fit(X, y)

clf.predict(X[:5])
```

Out[47]: array([0, 0, 0, 0, 0])

In [ ]:

In [ ]: